

AD-A146 013

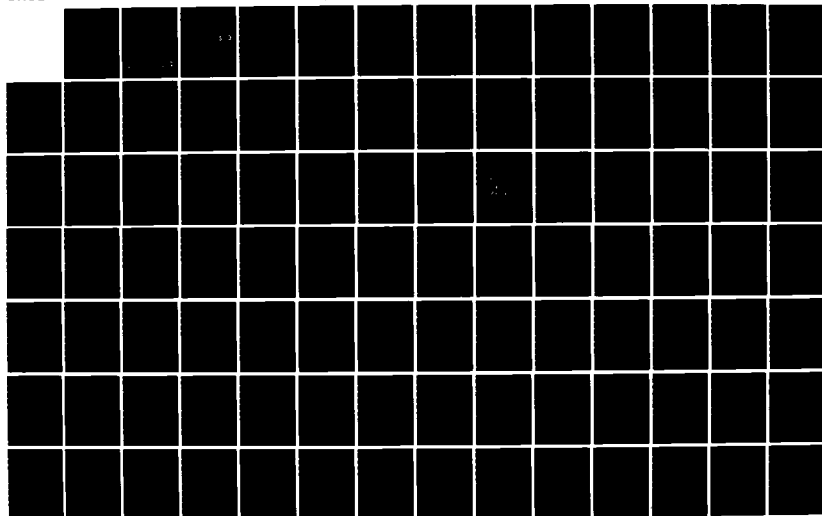
ANALYSIS OF THE TRANSPORTATION NETWORK FOR THE EXPORT
OF US STEAM COAL(U) MASSACHUSETTS INST OF TECH
CAMBRIDGE S R LINDBERG SEP 84

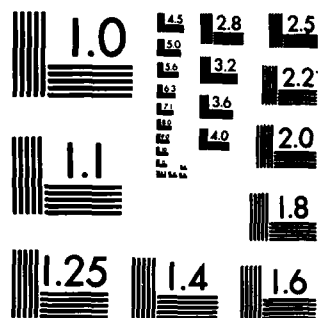
1/3

UNCLASSIFIED

F/G 5/3

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

AD-A146 013

DTIC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

DTIC
ELECTE
SEP 26 1984
S B D

84 09 10 091

ANALYSIS OF THE TRANSPORTATION NETWORK
FOR THE EXPORT OF US STEAM COAL

by

Steven R. Lindberg

B.S., UNITED STATES MILITARY ACADEMY
(1973)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF THE
DEGREE OF

MASTER OF SCIENCE
IN TRANSPORTATION

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 1984
© Steven R. Lindberg 1984

DTIC
ELECTE
SEP 26 1984
B

The author hereby grants to M.I.T. permission to reproduce and
to distribute copies of this thesis document in whole or in part.

Signature of Author _____
Department of Civil Engineering

Certified by _____
George A. Kocur
Thesis Supervisor

Accepted by _____
Amedeo Odoni
Chairman, MST Program Committee

Accepted by _____
Francois Morel
Chairman, Departmental Committee

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

ANALYSIS OF THE TRANSPORTATION NETWORK
FOR THE EXPORT OF US STEAM COAL

by

STEVEN R. LINDBERG

Submitted to the Center for Transportation Studies
on July 13, 1984 in partial fulfillment of the
requirements for the Degree of Master of Science in
Transportation

ABSTRACT

The steam coal trade boomed following the doubling of crude oil prices in 1979. The world was caught unprepared for the steep price climb in oil, and it was equally unprepared for the rapid increase in the coal trade. With two of the major world producers, Australia and Poland, unable to respond to the increased demand, the United States was able to gain a share of the increased coal trade. However the US was unable to respond quickly due to its limited coal transshipment facilities and its lack of large vessel capacity at its ports. The combination of increased coal demand and limited port facilities led to the planning of large scale port facility construction at many ports. The decisionmaking difficulties that followed has not yet been resolved because no agreement can be reached about where to employ the limited government resources available for port improvements.

This thesis develops a model of the entire export coal transportation network from the mine at the origin to the utility at the destination, to address the question of whether and where to dredge. By modeling the entire network, changes in any link can be examined to determine its impact on the network as a whole. In this study the impact of port dredging is investigated by modeling improvements at Baltimore, Hampton Roads, and Mobile. The result of the study is the clear indication that dredging Baltimore alone is the best option available now at low export coal volume, and that the benefits of dredging will increase as the volume of export steam coal increases. This solution is robust in that the changing of rail and vessel costs in a sensitivity analysis does not affect the solution materially. The US will still be the high-cost producer, but the dredging of Baltimore will improve the ability of the US steam coal producers to compete on the world market.

Thesis Supervisor: Professor George Kocur

Title: Assistant Professor of Civil Engineering

ACKNOWLEDGEMENTS

It is truly difficult to express appreciation to all the people who have aided and sustained me as I have worked on this thesis and throughout my year at MIT. I cannot acknowledge everyone here and so I will not attempt it. I only hope that they all know of my gratitude to them.

I do want to single out a few individuals who have been instrumental in the completion of this work. First Professor George Kocur who served as my thesis supervisor, and who was a constant source of aid and encouragement. Also to John Uppgren, who spent many hours writing an unfriendly menu-driven input routine so that this model could be user-friendly.

But most of all I want to thank my wife, Maribeth, and my three boys. For the entire year she took the main responsibility for raising our sons without complaint. At the same time she was a strong motivation and encouragement to me. I could not have accomplished this thesis without her continuous support.

TABLE of CONTENTS

TITLE PAGE	1
ABSTRACT	2
ACKNOWLEDGEMENTS	3
TABLE OF CONTENTS	4
LIST OF FIGURES	6
LIST OF TABLES	7
 CHAPTER 1: EXECUTIVE SUMMARY	 8
 CHAPTER 2: THE US POSITION IN THE WORLD COAL MARKET ..	 17
2.1 EXPORTERS	18
2.2 IMPORTERS	22
2.3 US COAL INDUSTRY	26
2.4 US COAL TRANSPORTATION SYSTEM	28
 CHAPTER 3: COMPUTER MODEL	 33
3.1 FLOW-DEPENDENT SHORTEST-PATH ALGORITHM	36
3.1.1 INPUTS BY USER	39
3.1.2 FLOW-DEPENDENT ALGORITHM OUTPUTS	42
3.2 TRANSSHIPMENT ALGORITHM	44
3.2.1 INPUTS BY USER	45
3.2.2 TRANSSHIPMENT ALGORITHM OUTPUTS	47
3.3 ALGORITHM COMPARISONS	49
 CHAPTER 4: METHODOLOGY and DESCRIPTION of the RUNS ...	 54
4.1 METHODOLOGY	56
4.2 BASE CASE	57
4.3 CHANGES FROM the BASE CASE	62
4.4 CASE STUDIES	65
4.4.1 BENCHMARK SCENARIO	65
4.4.2 DREDGE BALTIMORE	69
4.4.3 DREDGE HAMPTON ROADS	70
4.4.4 DREDGE MOBILE	71
4.4.5 DREDGE BALTIMORE and HAMPTON ROADS ...	72
4.4.6 DREDGE HAMPTON ROADS and MOBILE	73
4.4.7 DREDGE BALTIMORE and MOBILE	74
4.4.8 DREDGE BALTIMORE, HAMPTON ROADS and MOBILE	75
4.4.9 BASE CASE at 64 MTPY	79
4.4.10 DREDGE BALTIMORE	79
4.4.11 DREDGE HAMPTON ROADS	81
4.4.12 DREDGE MOBILE	82
4.4.13 DREDGE BALTIMORE and HAMPTON ROADS ...	83
4.4.14 DREDGE HAMPTON ROADS and MOBILE	84
4.4.15 DREDGE BALTIMORE and MOBILE	85
4.4.16 DREDGE BALTIMORE, HAMPTON ROADS and MOBILE	86

4.5	SUMMARY of the RUNS	87
4.6	SENSITIVITY ANALYSIS	90
4.6.1	SULFUR SURCHARGE	90
4.6.2	RAILROAD PRICES and NEW VESSEL RATES .	92
4.6.3	CONCLUSIONS	94
CHAPTER 5:	COMPUTER ISSUES	96
5.1	PROGRAM MAINTENANCE	96
5.2	PROGRAM SPEED	97
5.3	COMPUTER MEMORY	99
5.4	PROGRAM EXPANSION	100
BIBLIOGRAPHY	102
APPENDIX A:	PROGRAM OPERATING INSTRUCTIONS	104
APPENDIX B:	LINK COST COEFFICIENTS - FLOW-DEPENDENT ALGORITHM	107
APPENDIX C:	LINK COSTS	138
APPENDIX D:	FLOW DIAGRAMS and O-D COST MATRICES	148
APPENDIX E:	PROGRAM LISTINGS	209

5

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By PER LETTER	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



LIST OF FIGURES

FIGURE TITLE	PAGE
1. Export Coal Transportation Network in the United States	34
2. Export Coal Transportation Network in Foreign Countries	35
3. Example of the Shortest-Path Algorithm	37
4. Comparison of User and System Solutions	52

LIST of TABLES

Table Title	Page
1. Typical Link Costs	12
2. Port Dredging Costs and Savings	14
3. Port Flows	66

1. EXECUTIVE SUMMARY

When the oil price rise occurred in 1979, the entire world, and the US in particular, was unprepared for it in many ways. Plans had not been made to deal with such a crisis, and the reactions of many nations were quick and not well thought out. Many nations attempted to turn to coal as an alternate energy source, and a boom was begun in the world coal trade. However, two major coal exporters, Poland and Australia, were experiencing labor problems and were not available to help fill the increased demand for coal. In their absence the US was able to play an important role in the coal trade between 1979 and 1981.

However the US was not prepared to export the quantities of coal demanded in the time required because of inadequate port and transshipment facilities. Forecasts of the continuing boom in the coal trade, with an important role for the US, fostered plans for large-scale increases in the US port facilities to relieve the long lines of ships at US ports waiting to load coal. Every port had plans to join the coal trade, and the planned projects to increase coal transshipment facilities amounted to a capacity of three times the median forecast of the demand for US coal overseas. Thus began the political battle to gain government approval and funding of these new port construction and improvement projects.

In the event, the demand for coal did not keep pace with the forecasts. An oil glut appeared, and the price of oil fell. Where

there had been a great rush to change from oil energy to coal, the rush was slowed as the price differential narrowed between the two competing energy sources. In the US the export coal demand leveled off and then dropped. Some claimed that if the price of US coal could be lowered through cheaper, more efficient transportation, US exporters could increase their market share in the world markets. Thus there was still an effort made to improve the port facilities by dredging and by increasing the capacity of transshipment facilities.

The purpose of this study, and specifically of this computer model, is to build a model to evaluate the impact of the transportation system on the US export coal trade. By modeling the entire export coal transportation network from the mine at the origin to the final user at the destination, the effect of changes on any element of the system can be studied. Several important questions can be explored using this model. What effect will port dredging have on the delivered cost of coal? Which are the best ports to dredge economically and to what capacity should they be dredged? What policy of port and user fees should the government employ to recover their investments in dredging, assuming that the dredging is funded by the government? What effect will railroad deregulation have on the delivered cost of coal?

All of these questions concern only the initiatives taken to improve US coal exports. The important question is not so much the delivered cost of coal by itself, but how that delivered cost

compares with the competition. The US is now the high-cost producer of coal for export by a wide margin. It is highly unlikely that transportation improvements alone can change that fact. However the US has other advantages in the coal trade such as relatively stable labor-management relations and an excess capacity of production over domestic use. If the US cannot compete on price alone, can a price decrease combined with these other advantages be sufficient to improve the US market share? This qualitative question probably does not have a definitive answer that can be derived solely by a quantitative model. However the model can provide information on the ability of US exporters to trim the price differential with the competition, and it can provide the means to investigate different options of how to decrease the delivered cost.

The study in this thesis revolved around the dredging of ports on the US east and Gulf coasts to gain economies of scale on the ocean transportation links of the coal network. Dredging at the ports of Baltimore, Hampton Roads, and Mobile was modeled to determine the impact on the delivered costs of a ton of coal. Dredging of these ports was studied singly, in pairs, and all together to find the best possible combination. Each scenario was studied for a low volume of 32 million tons per year and for a high volume of 64 million tons per year of steam coal exports.

The delivered costs in this study ranged from \$52 - \$81, except for the costs to Madrid which were about \$15 higher from every US source due to the very long and expensive foreign rail link from the port

of Cadiz. The table of typical link costs (Table 1) shows that the mine production costs are the largest portion of the final cost by a large margin (58%). However the next most costly links are the US rail and the vessel links, each representing about 9% of the total delivered cost. The foreign rail link carried a relatively high cost as well (8%), but the remaining links did not individually contribute a large proportion of the total costs. It is important to note that the US port costs, which reflect the government user fees, only contribute about 3% of the total cost.

The only option which made any impact on the lowest deliverable cost of coal was to dredge the port of Baltimore. This result was fairly robust in that in every scenario in which Baltimore was dredged, the cost of coal to Central Europe was lowered by an average of \$2 per ton. The sensitivity of this solution to the amount of sulfur in the coal and to the use of railroad prices as opposed to costs was also examined. In both cases the advantage of dredging Baltimore was lessened, but it was still an improvement over the present export coal network with respect to delivered cost. When Hampton Roads and Mobile were dredged, they too decreased the cost of coal delivered to Europe through those deeper ports, but that cost was not as low as the delivered cost to Europe through other ports in the same scenario. As such these ports would not capture the trade from the shallower ports. While dredging these ports would improve local conditions, it would not improve system performance and cannot be justified economically.

TYPICAL LINK COSTS .

<u>LINK</u>	<u>COST</u>	<u>PER CENT</u>
MINE	26 - 42	58
TRUCK	1 - 2	3
RAIL	5 - 29	9
TRANSSHIP	1 - 3	3
PORT	0 - 2	2
VESSEL	5 - 17	9
FOREIGN PORT	1 - 2	2
FOREIGN TRANSSHIP	3	3
FOREIGN RAIL	4 - 16 (31)	8
UTILITY	2	3
<hr/>		
TOTAL	52 - 81 (96)	

All Costs in \$/ Short Ton

TABLE 1

When the port of Baltimore was dredged in scenario 2, the vessel costs to Central Europe were reduced by \$4 per ton, on average, while the port fee was raised from zero (\$0) to \$1. A 50% cost recovery by the government was used to calculate the port fees, which means that there is some subsidy to shippers contained in the dredging. When Hampton Roads was dredged in scenario 3, the vessel costs were reduced by \$2, and the port fees increased from \$0 to \$1. For Mobile the respective figures were a \$4 cost decrease and an increase of \$2 in the port fee. These figures for the port fees are almost the same as those predicted by the US House of Representatives in 1982. A full set of link costs for scenarios 1 and 2 are listed in Appendix C, as are the vessel costs both with and without dredging. One can readily see that the advantage of dredging Hampton Roads is almost totally negated by the rise in the port fees. At Mobile the relative advantage of dredging is about \$3 per ton, but the costs of the railroad network from the coal producing areas to Mobile are very high. Because of these rail costs, coal shipped through Mobile is still not competitive even when the port is dredged. Only at Baltimore, where there is a low-cost railroad network to the port, does the cost of the vessel link fall enough to overcome the rise in the port fees and allow the delivered cost of coal to be reduced.

Table 2 shows the dredging costs of each dredging option and the results of the one year savings on the total transportation system. From this table it is clear that only Baltimore presents a viable

PORT DREDGING COSTS and SAVINGS

<u>PORT</u>	<u>DREDGING COSTS</u>	<u>RUN COSTS</u>	<u>1 - YEAR SAVINGS</u>
BASE (32 mtpy)	0	1941	-
BALTIMORE	302	1917	24
HAMPTON ROADS	480	1930	11
MOBILE	407	1938	3
BALT & HR	782	1909	32
HR & MOB	887	1927	14
BALT & MOB	709	1914	27
ALL THREE	1189	1906	35
BASE II (64 mtpy)	0	4191	-
BALTIMORE	302	4142	49
HAMPTON ROADS	480	4175	16
MOBILE	407	4176	15
BALT & HR	782	4131	60
HR & MOB	887	4160	31
BALT & MOB	709	4127	64
ALL THREE	1189	4116	75

All Figures in \$ Millions

TABLE 2

alternative on a cost basis. In the case of Baltimore, there is a capital subsidy of \$151 million based on the 50% recovery of the dredging costs which is not included in the cost column. The interest on the \$302 million investment at any interest rate less than 8% is less than the \$24 million in yearly savings from this option. This relationship does not hold for either of the other ports studied in this thesis. The reader should note that as the volume increases, the financial advantage of dredging Baltimore also increases. The options in which Baltimore is dredged with another port appear to be feasible as well, but the effect on the savings is merely additive from the cases of dredging the individual ports. From this point of view, the additional dredging would only serve to reduce the relative savings which could be realized by dredging Baltimore alone.

Another important result of this study was to identify the Great Lakes port of Ashtabula, and possibly others on the Lakes, as an important export facility to small foreign ports where the advantages of large vessels cannot be employed. Being located close to the Pennsylvania and West Virginia coal fields allows coal shipped through Ashtabula to incur only small railroad transport costs, and thus to compete with those ports which are further from the coal fields. Based on this study, an investigation should be done in the longer term to determine whether the transshipment facilities at Ashtabula should be increased to gain more of this trade to small foreign ports. Dredging is not practical because of

the limitations imposed by the St. Lawrence Seaway, but increased transshipment facilities may allow Ashtabula to increase throughput and to compete for increased coal exports.

The dredging of Baltimore and increasing the transshipment facilities at Ashtabula will lower the cost of coal delivered overseas, but it will not lower that cost enough to compete directly on cost with South Africa or Poland. The advantage of these improvements, then, is a relative advantage. It does improve the US ability to compete on cost, but whether this improvement is good enough is unclear. The question of whether to make these improvements will have to be based on different criteria than just the delivered cost of coal to foreign markets.

2. THE US POSITION IN THE WORLD COAL MARKET

In the late 1970's and early 1980's, the world experienced an energy crisis in the form of a doubling of the price of crude oil by the OPEC countries. This was the second such energy crisis in less than 10 years, and it gave rise to strong fears that the price of oil might continue to climb at a rapid rate. In order to lessen their dependence on oil, many nations sought other forms of energy such as coal and nuclear. The coal energy was available sooner than was the nuclear, and the export market for steam coal exploded. Between 1979 and 1981 the US exports of steam coal jumped from 2.3 million tons to 30.0 million tons. This tremendous increase in coal exports caught the US unprepared, especially in the area of port loading facilities, and it sparked a great interest in building new facilities to support this export surge.

Prior to 1979 the majority of the coal exported by the US was metallurgical coal which was used in the making of coke for the steel industry. The majority of this coal was shipped from northern and central Appalachia to a few steel producing areas which were near large ports. This trade was well developed, and the exports were rising gradually in the late 1970's. The export steam coal trade at that time was mainly with eastern Canada, and there was a very small overseas component. With the OPEC oil price rise in 1979, steam coal exports rose to the level of

metallurgical coal and promised to surpass it. However there was already a competitive environment in the world coal market, and the US exporters did not enter that environment with many advantages.

2.1 EXPORTERS

The main exporters of steam coal are South Africa, Australia, Poland, Canada, and the US. Until 1979 most of the steam coal trade was between neighboring countries, but the explosion in demand ended that and caused greatly increased shipments by the ocean trading lanes. This should have increased competition tremendously, but several of the exporters faced serious production and transportation problems in 1979. This opened the door for higher-cost producers to enter the export coal market. Since 1981 there has been a weakening in the demand for coal on the world market as the price of oil dropped, and as some of the traditional coal exporters controlled their problems and resumed exporting.

South Africa has a strong infrastructure in place for exporting coal, and an exceptionally strong position in the European market as the low-cost producer. Coal is the main source of energy in South Africa, and it is maintained at a below-market price internally by the government. The South Africans export coal to

subsidize their internal needs and to gain hard-currency foreign exchange. The government strictly controls exports by linking export licenses to internal production at the subsidized prices, and by controlling the transportation system including the railroads and the ports. The government extracts very high user fees for the use of its transportation facilities, but the extremely low production costs still enable the South Africans to deliver coal to Europe cheaper than any other exporter.

The racial tension in South Africa causes many would-be importers to limit their imports because of the possibility of a shutdown at the mines. In fact there are several nations, such as Sweden, who will not trade with South Africa at all because of their racial discrimination policies. At the same time the South Africans tend to limit their own exports, and to allow higher-cost producers to gain market share. By letting someone else set the market clearing price, they are able to charge just below that price and make very high profits on the difference. The government encourages this behavior strongly with its export license control. South Africa should be able to maintain its strong position in the European market barring any racial/labor problems.

Poland is the only exporting nation which can challenge the South African delivered price of coal to Western Europe. When the last oil crisis hit in 1979, Poland was not in a position to export coal because of the labor unrest over their union Solidarity and the ensuing martial law which destroyed the union. The Poles had

been a major supplier to Western Europe before that period, and they have regained some of their position since 1982. Proximity is a major advantage of Poland in this market, with the resulting low transportation costs. The Polish economy is centrally directed, and the coal industry and the transportation network are very tightly controlled. The instability of the labor-government relations is still just below the surface, and it weighs heavily on importers. However, many of the importers are also creditors of Poland, and they are anxious for the Polish economy to regain its health so that the Poles can repay their debts. With its price position and the feelings of its creditors, Poland is in a position to be a force in the European coal market.

The Australian coal industry is able to compete in both the European and the Pacific markets, but it is in the Pacific Ocean area that they are the low-cost producer. Australia's big advantages in the Pacific are proximity and rich coal seams near their major ports. Here, as in South Africa, the government extracts heavy user fees from users of the railroads and ports. In addition the Australian government has discouraged outside investment by mandating that at least 50% of any foreign investments in Australia be financed in Australia. This law has slowed the development of their coal industry along with the rest of their economy. This deliberate policy of slow development has allowed higher-cost producers to capture some of the market in the Pacific. And, as with South Africa in Europe, it has allowed the

Australians to gain high economic rents by pricing just below their higher-priced competitors.

The Australians are not viewed as a highly reliable trading partner because of the strength of their unions. Especially in their port workers' unions, a strike by a small number of workers is able to close down large segments of the transportation infrastructure. In fact such a strike occurred in 1979 when the boom in the world coal trade began. This strike allowed other exporters, especially the US and Canada, to enter the Pacific markets. Though Australia has regained much of its market, it is still viewed warily by importing nations.

The other major coal exporter, other than the US, is Canada. Its exports come mainly from western Canada and flow to Pacific Ocean markets. Eastern Canada imports coal from the eastern coal fields of the US. The Canadian fields are located in the Canadian Rocky Mountains, and many of the seams are very hard to work and to transport from because of the terrain. The port of Vancouver is prepared to export large amounts of coal, and a transportation infrastructure is in place and improving. This will give Canada the ability to export increasing amounts of coal, though it will remain one of the high-cost producers.

There are many other nations which mine coal, but most do not mine enough to fill domestic requirements and are net importers. China, Columbia, and India are recognized as potential exporters in the future, but all will require extensive investment before

that will occur. In each case not only do the coal mines have to be developed, but the transportation infrastructure of railroads and ports also will have to be constructed to support exports.

The Soviet Union contains almost half of the world's known coal reserves, but most of it is located in economically and climatically inaccessible areas. Those reserves which are in the western portions of the country have been seriously depleted to the point where they cannot support large scale exports. The Soviet Union will probably continue to export small amounts to other Eastern European countries, but it will not be a major force in the world market.

2.2 IMPORTERS

The major importers in the coal market are the nations of Western Europe and the Pacific rim countries, especially Japan. Most of these countries also lack supplies of oil, and they are very sensitive to price rises in their energy sources. When the OPEC nations doubled the price of oil in 1979, these importing nations immediately began to look for alternate sources of energy. Coal trade on the world market recorded the tremendous increases as a result.

Prior to 1979 the majority of coal traded on the world market was metallurgical coal used in coke production in the steel industry.

This trade tended to be concentrated in a few areas such as the Ruhr area of Germany and the Taranto area of Italy. When steam coal became a force in international trade, the nature of the trading changed. Steam coal is used predominantly to produce heat for electric power. The demand for such energy is very widespread, touching every continent and almost every nation in the world. Because steam coal is less valuable than metallurgical coal, the cost of transportation represents a much greater share of the final cost. This necessitates shipping more directly to the final user, further complicating the trading of steam coal.

Many nations are trying to gain economies of scale by building port facilities capable of handling large vessels. France is one of these nations, and it is currently increasing its facilities at Dunkirk, Le Havre, and Marseille. With large port facilities comes concentration of imports and increased internal transportation costs to deliver the coal to widely dispersed final users. France has a highly centralized economy, and the import of coal is handled by one agency. Such an organization should allow France to order large quantities at good prices. The French have also determined that diversification of suppliers is in their best interest, and they have set a limit on the quantity of coal that they will import from any one supplies. France is attacking the import of coal and oil in a different way as well, by vastly increasing its use of nuclear energy. If all the nuclear development that France is planning becomes reality, that country

may become an atypical coal user with more use by general industry than by the electric utilities. It is projected that France will remain a net importer of coal, with a stable market size.

West Germany and the United Kingdom both mine a good deal of coal within their own borders, but both are still net importers. In both countries internal production is subsidized, and there are import limits which also support local production. The importing organizations are decentralized in both nations, although in Great Britain there is a Central Electric Generating Board which operates nearly all the generating capacity in England and Wales. Neither West Germany nor the United Kingdom has any major coal port facilities which can handle large coal vessels.

Interestingly, they both utilize the port of Rotterdam, Netherlands, and transship their coal by barge to gain economies of scale for the ocean transport costs. Both countries are projected to increase their imports of steam coal in future years.

Japan is the largest importer of coal in the Pacific area, and its imports are forecast to rise swiftly as they try to reduce their dependence on imported oil. Japan is a maritime trading nation, and it has many large ports capable of handling supercolliers. Being an island nation with vast coast lines compared to its land area, the inland transportation is less important to Japan than is the ocean transportation. The use of large vessels to capture the economies of scale, then, is very important to gaining exports to Japan.

Most other importers are characterized by small demand quantities scattered locations, and small port facilities. Such importers are numerous in Europe, northern Africa, and the Pacific rim. In the aggregate they constitute a considerable market, but it is a market which is difficult to categorize and to supply efficiently. Transportation costs will tend to be higher to these nations because of the limitations of their transportation infrastructure. These nations also tend to be Third World or poorer nations with highly volatile economies. The future exports to these countries is uncertain, but there is potential for them to be significant.

2.3 US COAL INDUSTRY

The coal mining industry in the US is very heterogeneous, but it can be broken into two main coal producing regions. The Appalachian coal fields have long been mined for domestic coal consumption, and they are currently supplying the majority of the export coal trade. These fields range from Pennsylvania and West Virginia in the north to Alabama in the south, and they export coal through ports on the Atlantic Ocean, the Gulf of Mexico, and the Great Lakes. The western coal fields are much newer, and they employ more surface mining than do the Appalachian fields. One of the driving influences in these western fields is the increasing use of imported coal by the Pacific rim countries, and the possibilities of exporting there through the west coast ports of the US.

The Appalachian fields range in size from family owned mines to large corporation owned mines. About half of the production there comes from relatively small mines causing consolidation for shipment to be an issue. The Appalachian coal is characterized by a high heat content and a high sulfur content. The first characteristic is beneficial since the higher heat content reduces the quantity of coal which must be transported to meet user needs, and reduces the impact of the transportation costs. Sulfur, however, is a pollutant, and a maximum allowable level is often specified in contracts. Excess sulfur can require users to

install costly scrubbers on their plants to meet environmental standards, and many importers are not willing to make that investment.

The United Mine Workers Union is very strong in Appalachia, and most of the large mines are of the deep shaft variety. Both of these factors tend to keep extraction costs high. The labor-management relations in the US are seen to be among the most reliable in the world, however, because they are predictable. Coal importers can be quite certain that there will be no major labor problems unless a contract is nearing its expiration. These times are well defined, and importers can make arrangements well in advance to avoid potential problems.

The transportation infrastructure supporting the Appalachian coal mines was created to support the domestic coal markets. It is well established, and it can be fairly efficient. The system has some slack capacity, and it is capable of meeting some increased demands. For the typical mine the links in the transportation system are mine production, minemouth to tippie, tippie to port, transshipment at port, ocean voyage by vessel, and the reverse links at the importing end. These elements will be discussed in more detail following a brief discussion of the western coal fields.

The western coal fields have not been developed for as long nor to the extent of the Appalachian fields, and they tend to produce coal at a lower cost. Their sulfur content is also lower, but the

western coals tend to have lower heat content. They are located far from the western ports and require a long domestic transportation link. For these reasons, western coal tends to cost more per unit of heat when delivered in Europe than does Appalachian coal, although there has been some evidence that western coal can compete in the Pacific markets as the high-cost producer. Another disadvantage of western coal is the relatively underdeveloped transportation network in the coal region. Whereas in Appalachia nearly every mine is within 10 miles of a tipple, in western areas this density has not yet been established. This causes the movement from the mine to the tipple to be relatively expensive (either by truck or by rail due to construction costs), and it causes sharp increases in the total transportation costs. The ability of the western coal region to compete in the Pacific should improve as these fields are developed and as the transportation network is improved.

2.4 US COAL TRANSPORTATION SYSTEM

Once coal is removed from the ground, it must be transported to an ocean port and loaded on a vessel for export. This section will describe this system in the US as a series of links. In general these links include movement from the mine to a long-haul loading site, long-haul transport to a port, transshipment from

the long-haul carrier to the vessel, and the voyage of the vessel to the foreign port. At that end the sequence is reversed for delivery to the final user. In addition to these links there is some cost for extraction, and there may be charges by the government for the use of facilities such as the locks and dams on a river system or user fees for port facilities. This description is very general since there may be more than one means of transporting the coal over a given link. The choices are generally limited by the location of the mine and the available modes as much as by price differences.

The first link in this transportation system, from the mine mouth to the tipple, is often the most expensive on a per-mile basis of the entire system. This distance is generally as short as possible and is often accomplished by truck. Trucking coal is very expensive, which is why mine operators try to keep the distance short. Producers suggest that increasing the capacity of trucks by allowing larger trucks or more trailers on trucks may be a way to keep line-haul modes competitive. The argument is that larger capacity trucks will allow operators more flexibility in the tipple they choose and cause price competition among the line-haul carriers. In the absence of increased truck user fees and if there are additional tipples close to individual mines, this may be possible. Some of the larger mine operators are able to reduce the cost of this link by constructing conveyors to the tipple or by inducing the line-haul carriers to extend service to

the mine. In an industry where half the volume is produced by small operators, these alternate means are clearly the exceptions rather than the rule.

There are two efficient line-haul vehicles for coal - rail and barge. The choice between the two is dependent more on mine location than it is on price. Both modes stress the same characteristics to gain their economies - long distance haul of bulk quantities by a small crew. The barge traffic from Appalachia terminates mainly in New Orleans and Mobile. It usually consists of several large barges lashed together into one unit and propelled by a single river tugboat. Rates are generally quoted by the ton or by the barge.

Coal trains rarely carry any other cargo. In fact there is a good deal of specialized equipment being developed to haul coal, from special types of cars to cars made from special materials. The basic rate to transport coal is by the car, but multiple car shipments and unit trains may bring reduced rates to the shipper. With railroad deregulation there has also been a move to contract rates. By signing a long-term contract, the shipper can get reduced rates and improved service. The railroad gets a fixed shipment amount which allows it to standardize operations and to reduce its operating costs.

Once the coal arrives at the port it must be offloaded from the line-haul vehicle and loaded on the vessel. This may be a one step operation of direct transfer, but more likely it will entail

some storage or inventory between the two functions. This transfer step is called transshipment. When the boom hit the US in 1979, there were not enough transshipment facilities in the country. The result was long queues of ships awaiting loading. Being the high-cost producer, the US exporters could not afford the extra cost of demurrage, or vessel wait time in the ports. Many ports immediately planned and began construction of new facilities to reduce demurrage. By some estimates the amount of planned capacity is three times the median of the forecast demand for US export coal.

There has been a lot written about the importance of vessel size on the delivered price of coal. On a per-ton basis, there is no doubt that larger vessels are much more cost efficient. However the US currently has no capability on the east coast to load ships of over 100,000 deadweight tons. All of our major competitors have this capability, and many of the importers, especially the large importers, have this capability as well. The lack of superports clearly impacts on the delivered price of coal, and there have been proposals to dredge almost every major east coast port to gain this capability. The question is now political because almost all the proposals require funding by the Federal Government. Every port wants the business that increased capacity and large vessel economies would bring.

The Federal Government is also exploring ways to recapture their investment in dredging by imposing user fees at the ports. What

form should these fees take, and who will they affect? Will they negate the advantages gained by dredging the harbors? Given the decision to dredge certain ports with a particular user fee policy, there will be gainers and losers from the present environment, and the political battle to be a gainer will be fierce.

The transportation system is nearly the same at the foreign end of the ocean shipment. There is another transshipment facility to move the coal from the vessel to the vehicle for inland movement. There may be port fees and user charges. Inland transportation may be either train or barge depending on location, and there will probably be another short-haul movement by truck to the final user. There are two major differences between the US side and the foreign side transportation systems. First the US has no influence over the operation of the foreign system, and, hence, cannot improve that system to increase the efficiency of the US coal exports. Second much less is known about the foreign system in the way of rates and fees. The latter factor is important in the estimates of the delivered price of US coal, and it must be kept in mind as a limitation of the results obtained.

3. COMPUTER MODEL

The approach that I have taken to determine the impact of changes in the transportation system on the delivered cost of coal is to model the system as a network. This network is composed of all the links from the mining of the coal at the origin to its arrival at the destination utility (see Figures 1 & 2). By modeling the entire network, great flexibility is given to the user to change any transportation link in order to determine its impact on the system as a whole.

Once the network is established with the construction of the individual links and link costs, this model will develop the least costly paths from every origin to every destination. This is accomplished with a flow-dependent shortest-path algorithm which allows the link costs per ton to vary as the flow on the link changes. Once the equilibrium link flows and costs are established, the model uses a so-called transshipment algorithm to determine the best origin from which each destination should buy coal to minimize its total cost, assuming constant link costs.

The remainder of this chapter will describe these two algorithms. First the flow-dependent algorithm will be presented. I will briefly describe how the algorithm works, what inputs are required of the user, and what outputs the user will receive. The same treatment will then be given to the transshipment algorithm.

Export Coal Transportation Network in the United States

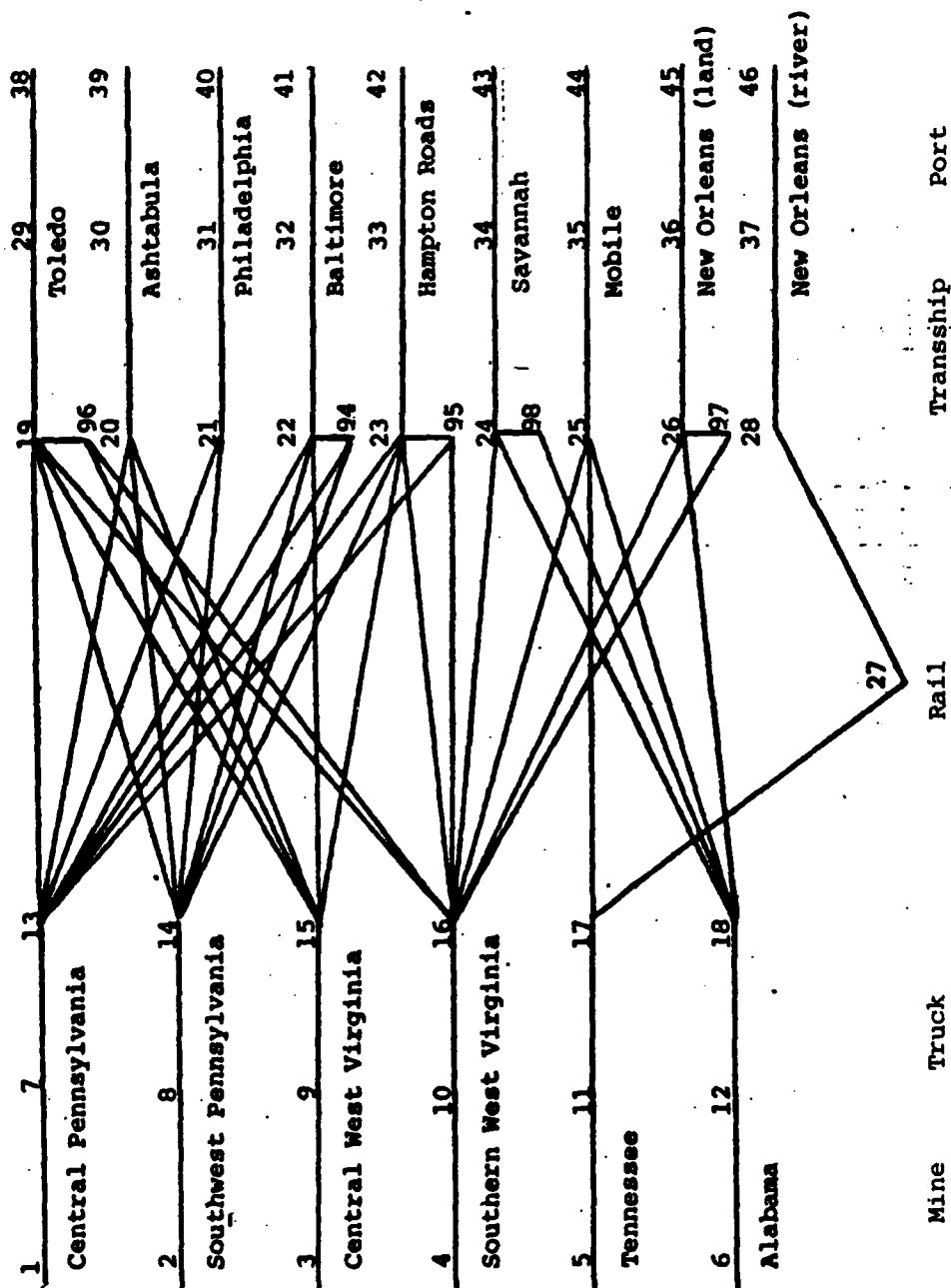


FIGURE 1

Export Coal Transportation Network in Foreign Countries

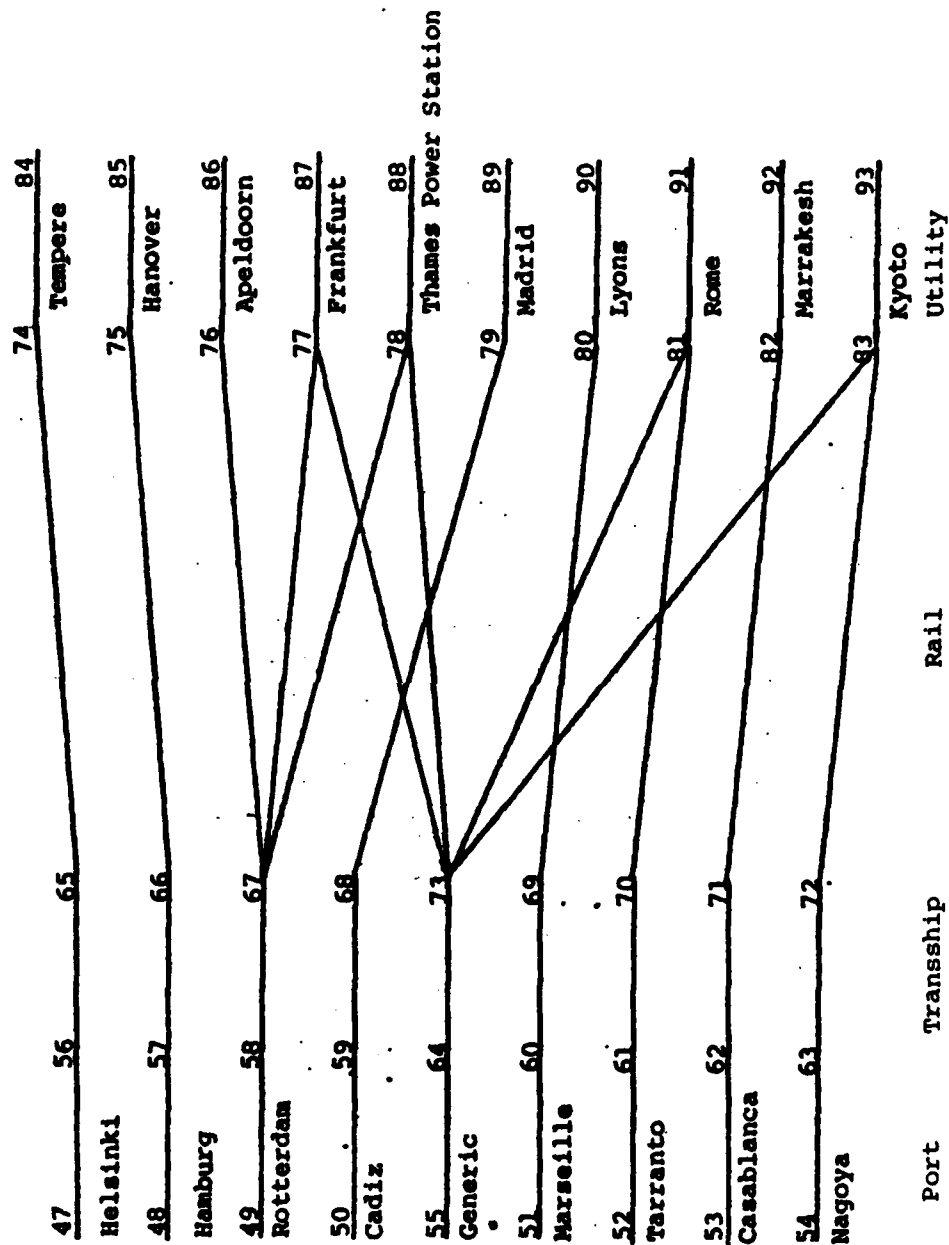


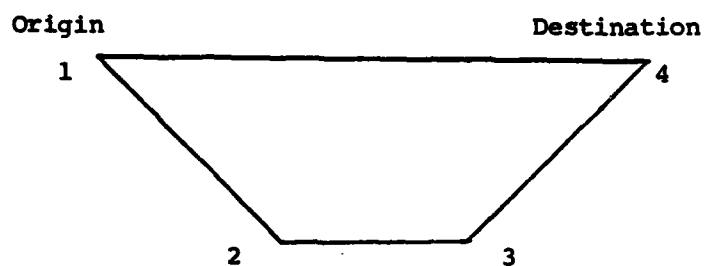
FIGURE 2

3.1 FLOW-DEPENDENT SHORTEST-PATH ALGORITHM

The purpose of the flow-dependent algorithm is to find the lowest cost path from every origin mine to every destination utility over a network on which the link costs change with changes in the link flows. The lowest cost paths are found using a variant of the shortest path algorithm from Horowitz and Sahni (2). Their algorithm has been expanded in this model to generate the lowest cost path for every origin-to-destination (O-D) pair.

As an example of the lowest cost shortest path algorithm, consider the network in figure 3. This network has an origin at node 1 and a destination at node 4. There are two paths from node 1 to node 4, and there are a total of four links which have constant costs as described on figure 3. The flow traveling from node 1 to node 4 will begin its search for the cheapest path at node 1. The shortest path algorithm keeps an array of the cost from the origin called a labeling array. The value for the origin is zero (0), and the value for any other node is the cost from the origin along the lowest cost path to that other node. When the network in figure 3 is initially inspected, the label for node 4 equals 4, and the label for node 2 equals 1. On the next search, which is done from node 2, the label for node 3 is assigned a value of 2. Finally on the search from node 3, the lowest cost path to node 4 is identified, and the label is changed to a value of 3. While

Example of the Shortest-Path Algorithm



ONODE	DNODE	COST
1	2	1
1	4	4
2	3	1
3	4	1

FIGURE 3

the search for the shortest path is in progress, the path to each node along the shortest path is retained by means of an array of the predecessor for each node investigated. The predecessor of a node A is that node from which node A is reached when using the shortest path from the origin. When the destination is reached, the entire shortest path can be traced back to the origin by means of this predecessor array.

Often the shortest path algorithm is run on a network with fixed link costs, but the flow-dependent algorithm better describes how the link costs behave on a real network. In this algorithm, as the amount of coal shipped on a link changes, the cost of shipping a ton of coal on that link also changes, usually increasing as the amount of flow increases. This is because increased flow usually leads to congestion and delay. However some links will experience economies of scale, and their average costs will decrease with an increase in flow. For example, transshipment facilities can load coal on larger ships more efficiently than on smaller ships since there is less non-productive time of docking and undocking. The result is lower average costs at transshipment facilities which handle larger vessels.

Since the shortest path algorithm runs only for fixed link costs and the link costs change with the link flow, some method must be employed to combine the changing costs and the fixed cost algorithm. This method is called the Convex Combinations Method by Sheffi (6). It is an iterative process whereby the shortest

path algorithm is employed with fixed link costs in successive iterations. Between iterations the link costs are changed to reflect the new flow pattern on the network. In order to avoid oscillating between two extreme solutions, the flow pattern from all previous iterations is combined in a weighted average to form the new flow pattern. This process will continue either a specified number of iterations or until the change in the flow pattern is sufficiently small to meet the user's requirements. At this point the network is said to be in equilibrium.

One assumption in this model is that the link flows and costs will converge to equilibrium. Sheffi proves that this is true when the network is convex. The network will definitely be convex, and hence will converge, when every link exhibits the normal behavior of congestion and diseconomies of scale. Convergence is less certain when some links exhibit economies of scale since the cost functions on those links are concave. If the impact of the concave links is small, the system will probably be convex overall and so converge. This has been the experience in test networks to this point.

3.1.1 INPUTS BY THE USER

In order for the user to run the flow-dependent algorithm, he must input information about the network, the link costs and the amount

of flow. In addition there are two control variables which the user can change from their default values if he desires. These variables are all input through a series of menus, described in Appendix B. This section summarizes the key items.

The network over which the flow-dependent algorithm is run is a directed network of links. Each link carries flow in only one direction, although opposite links to portray two-directional flows are allowed. Each link connects a head node (onode) and a tail node (dnode) which designates the direction of flow (see Figure 3). The user enters this network structure as part of the input menus. Care must be taken so that the dnode from one menu becomes the onode in the menu for the next link. This will insure that the network is connected from origin to destination.

The input for the link costs is entered indirectly by the user in terms of more basic components. The user will enter, for instance, the type of track and signal system for a railroad link and the length of the link, and the input program will compute from internal tables the parameters of the link cost equation, $COST = C0 + C1 * (FLOW) ** C2$. This equation holds for every link in the network, each having different parameters, $C0$, $C1$, and $C2$, based on the user inputs. The parameters will remain constant throughout the execution of the algorithm, and the cost will vary only with the flow. See Appendix B for a more complete discussion of the cost parameters.

The flow inputs by the user are origin and destination specific.

An entry is required for the flow from each origin mine region to each destination utility country, although zero flow entries are accepted and are the default. The algorithm will then distribute these flows to the links along the lowest cost O-D paths. Some of the links will appear on more than one O-D path, and those flows will be combined to determine the link cost.

The control parameters which determine the accuracy of the equilibrium are an optional input for the user. The first of these parameters is the number of iterations, which determines the number of updates in the link flows based on new link costs. The default value is 2 iterations, but the algorithm always executes one more iteration as it generates its output so that three iterations are actually performed when the default is used. The network which I used to test the export coal system consisted of 98 nodes and 191 links. On that network, the flow-dependent algorithm written in UCSD Fortran takes about 7-8 minutes per iteration on an Apple II. The user can change the number of iterations by changing the entry for iterations on the parameter menu during his input of data.

The second control parameter is a measure of the degree of convergence desired by the user, and it is called the convergence criterion on the parameter input menu. The algorithm uses a root mean square convergence test with a value called TOTAL. TOTAL is computed as a measure of the consistency of the flows between successive iterations. At each iteration the flow on each link is

compared with the link flow from the previous iteration. The difference is squared and then these values are summed over all the links. The square root of this sum is divided by the total flow on all the links of the network, resulting in TOTAL. This value is compared to the convergence criterion at each iteration to decide if the network is in equilibrium. The flow-dependent algorithm will terminate at the first exit criterion it meets, if either the number of iterations is completed or the value of TOTAL is less than the convergence criterion.

3.1.2 FLOW-DEPENDENT ALGORITHM OUTPUTS

Once the input has been entered and the flow-dependent algorithm has been executed, the user is interested in output. The output from the flow-dependent algorithm is a matrix of path costs for every O-D pair and a listing of all the links giving their costs and flows.

The O-D cost matrix is output from the flow-dependent algorithm proper. The algorithm executes for only one origin at a time, and the output to the O-D matrix is written for the one origin to every destination before the algorithm proceeds to the next origin. The costs entered are the path costs from the origin mine to the destination utility, but the matrix does not describe the individual links in the paths or the path routes. (These are

obtained as described below. See Appendix D, Figure 4.4.1C for an example.)

The link outputs of cost and flow follow immediately after the O-D cost matrix. This output gives the onode and dnode of every link in the network together with the link costs and flows. Every link in the network is listed whether there is flow on the link or not. Many links will have zero flow which means that they are not on the lowest-cost path of any O-D pair.

To determine the path for an individual O-D pair, there is a special subroutine. This subroutine uses the same shortest path algorithm, but it is executed only after the equilibrium link costs have been determined. After the link output from both the flow-dependent and the transshipment algorithms, the user will be prompted on the screen whether he wishes to examine the path for a specific O-D pair. If he answers 'y', the screen will prompt him for the origin and then the destination he desires. The user will enter the node number of the origin mine and the destination utility at the keyboard. The values will appear both on the screen and on the printout. The output from this routine will include the origin number, the destination number, the path cost, and the node number for every intermediate node in order. This single path routine will repeat as many times as the user desires until he answers 'n' when asked if he has another path to investigate.

3.2 TRANSSHIPMENT ALGORITHM

Given a directed, capacitated network with supplies at the origins and demands at the destinations, the transshipment algorithm will determine the flow pattern which results in the lowest system cost. The main task of the transshipment algorithm is to determine the O-D flow pattern across the network of links. This is valuable to the importers since it identifies their cheapest source of supply. It is also useful for US policy purposes, because it simulates what might happen as a consequence of possible changes in the US transportation network such as dredging of harbors and the deregulation of the railroads.

The transshipment algorithm is a derivative of linear programming . The transshipment algorithm works over a network of links. Thus not only is the cheapest source of supply identified for each destination, but the path from that supplier to the destination is also determined.

As with many linear programming formulations, the transshipment algorithm assumes that the link costs do not vary with the amount of flow on the link. The transshipment algorithm is run after the flow-dependent algorithm in order to take advantage of the equilibrium link costs which are an output of the flow-dependent algorithm. If the link flows on the transshipment algorithm are not much different than those on the flow-dependent algorithm, the fixed link costs assumption is probably acceptable. If there is a

great difference in the flows, there may also be a great difference in some of the link costs based on those link flows. The fixed costs assumption may not be valid in that case. In that case, the user must decide whether the flow pattern used by the flow-dependent algorithm is a good model of the actual flows of export steam coal. If the pattern is not a good model, the user must change the flow pattern appropriately. If the flow pattern is correct, the transshipment algorithm results suggest that there is a better solution available to the utilities about where to buy their coal. The user must understand that the transshipment algorithm leads to a system solution which may not be the best solution for each individual participant. This will be explained more fully in section 3.3.

3.2.1 INPUTS BY USER

The types of inputs required by the transshipment algorithm are the same as for the flow-dependent algorithm. The transshipment algorithm requires information about the network structure, the link costs, and the O-D flows. Although some of the inputs are slightly different than for the flow-dependent algorithm. These variables are entered on the same set of menus described in Appendix B.

The network structure of links and nodes is the same for the

transshipment algorithm as it was for the flow-dependent algorithm. However in addition to specifying the links, each link must be given a capacity which is the maximum amount of flow it can handle. The transshipment algorithm will assign all the the flow up to capacity to the lowest cost links. To ensure that the flow does not exceed that which the link can handle, the user will set the link capacity in the input menu as he enters the link into the network. The capacity should be set at the point where the link begins to experience unreasonable cost increases due to congestion for links with diseconomies of scale. For links with economies of scale the capacity should be set at the practical capacity of the link, for example, the practical capacity of the transshipment equipment at a port.

The link costs required by the transshipment algorithm are fixed costs on each link. At present these costs are the output equilibrium link costs from the flow-dependent algorithm. The formulation of these costs has already been described, and the use of these costs simplifies the data collection task to only the basic cost components and not to the final shipper costs themselves. However, if the flows generated by the transshipment algorithm are much different than the equilibrium flows from the flow-dependent algorithm, these costs will be suspect. If the user wishes to enter his own fixed costs, he is able to do so by entering his costs in the input menu in the column following the destination node for each link. These entries take precedence

over other cost inputs and will remain fixed throughout the program.

The origin to destination flow pattern is the output of the transshipment algorithm. Therefore the algorithm ignores the O-D flow matrix input by the user for the flow-dependent algorithm. The requirements for the transshipment algorithm are only the supply available at each origin mine and the demand required at each destination utility. There is no requirement for the sum of the supplies to equal the sum of the demands. The user enters these supplies and demands on the menus when he enters the mine production links and the foreign utility links. For the mine production links the production rate should be entered as the annual production in millions of tons. Similarly the rate of use of the foreign utility should be entered in millions of tons on the foreign utility menu.

3.2.2 TRANSSHIPMENT ALGORITHM OUTPUT

After the transshipment algorithm has been run, the output gives the flow pattern which minimizes the system cost when each utility is able to buy coal from any supplier. When the flow pattern is plotted on the network, the best source of supply for each utility is determined along with the overall flow pattern.

The output itself includes a listing of all the links which form

the basis, or solution set, of the transshipment algorithm. This will include at least one link to every node in the network, although some of these links will carry zero flow. Links at capacity are listed separately at the end of the basis solution, and are separately labeled. If there are no capacitated links in the network, there will be a message to the user to so inform him. If the total supply and total demand quantities are not equal, there will be some flow assigned to so-called artificial links which form the initial solution to the transshipment algorithm. These links are identified by one node which is not in the user's network and the other node which is a supply or demand node. These links should be ignored when the solution is applied to the network diagram.

There will be no link cost outputs since the costs are constant and the same as were output for the flow-dependent algorithm. The paths of the O-D flows are also not output in the transshipment algorithm. Immediately following this algorithm output, the user will be prompted about investigating individual O-D flows. This path routine has already been discussed, and it is important to remember that the path routine is written for the shortest path without considering link capacities. When it is run it will indicate the lowest-cost path for the O-D flow, but that path may not be possible, given the link capacities and the interactions of different O-D flows. The single path routine will, however, indicate the desired path for an O-D flow, and it can be useful to

investigate which links suffer congestion and what adjustments are made to those links.

3.3 ALGORITHM COMPARISONS

Both of the algorithms used in this program have strengths and weaknesses in determining the optimum flow pattern for the network. The two algorithms use different criteria to arrive at a solution. The flow-dependent algorithm computes the shortest path between the origin and the destination. As such it looks for a solution from the user's point of view. The transshipment algorithm computes the solution from a system point of view while determining the best supplier for each utility demand.

The strength of the flow-dependent algorithm is that it models the link costs in a dynamic setting where they change based on the link flow. This is the way most link costs really work, and it provides a dynamic solution. The other advantage of the flow-dependent algorithm is that a solution from the user point of view is the one most common in the free market. Every participant is trying to minimize his own costs, and there is little concern for the good of other network users. The biggest weakness of the flow-dependent algorithm is that the O-D flows must be predicted or modeled prior to running the algorithm, and that they are fixed outside the model. The market does not play a role in choosing

the best supplier for each destination in the algorithm.

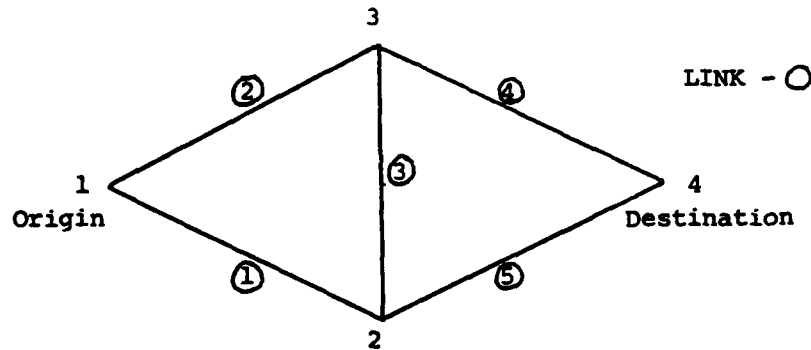
The transshipment algorithm complements the flow-dependent algorithm nicely by casting its greatest strength on the weakness of the flow-dependent algorithm. In the transshipment algorithm, the source of supply for each utility demand is determined by the algorithm. In many cases this sourcing may differ from the flow pattern input to the flow-dependent algorithm. Of course with some of the importing nations limiting the amount of coal they will import from any one supplier, the flow-dependent algorithm may model reality without resulting in the most efficient flow pattern. In fact the user will notice that the total system cost of the transshipment algorithm is always lower than the total system cost of the flow-dependent algorithm.

The weaknesses of the transshipment algorithm are that the link costs are fixed and that the system solution can only be achieved with central direction. The fixed link costs are a weakness because real-world transportation network link costs are not fixed and constant. As has been explained, the approximation of fixed link costs may be reasonable if the link flows are not much different in the transshipment algorithm than in the flow-dependent algorithm. The idea of the central direction is foreign to a free market, and it usually does not serve the individual to best advantage. In a system solution, some users give up a little of his own advantage, usually in the form of higher personal cost, to make the system as a whole more

efficient. It is plain to see that many users will have an incentive to change their path choices to gain better individual results. The individual will be better off in that manner, but the system will suffer. In a free market, the system solution is rarely seen.

As an example of the difference between the system and the user solutions, consider the network in figure 4. The five links serve one origin-to-destination pair, and they form three paths. Assume that a total of 6 flow units must travel from node 1 to node 4. The two solutions are given below figure 4, detailing the link flows and costs, the path costs and the total flow-cost units of each solution. Note that the quantity of flow-cost units and the path costs are lower in the system solution than in the user solution. One would think that the system solution is the one which the network would exhibit. However, the system solution is unstable because there is an incentive for an individual flow unit to move from path 1 or path 2 to path 3 to reduce its individual cost. The unit which moves would have a path cost of 81, which is lower than the 83 on the original path in the system solution, but in so doing the cost on the other path would increase to 93. The cost on his original path would fall to 82. At this point, one unit would move from the other original path, which had become the high cost path at 93, to path 3, and the user solution would be attained. In this case every user would be worse off than if each had accepted the system solution. The problem is that in a free

Comparison of User and System Solutions



LINK	ONODE	DNODE	UNIT COST (X = flow)	PATH
1	1	2	10X	1 1 - 2 - 4
2	1	3	50 + X	2 1 - 3 - 4
3	2	3	10 + X	3 1 - 2 - 3 - 4
4	3	4	10X	
5	2	4	50 + X	

USER SOLUTION

LINK	FLOW	COST	FLOW*COST
1	4	40	160
2	2	52	104
3	2	12	24
4	4	40	160
5	2	52	104
			552

SYSTEM SOLUTION

FLOW	COST	FLOW*COST
3	30	90
3	53	159
0	10	0
3	30	90
3	53	159
		498

PATH	COST
1	92
2	92
3	92

PATH	COST
1	83
2	83
3	83

FIGURE 4

economy there is no central directing authority to convince the users that they would be better off with the system solution.

The flow-dependent algorithm works from the user solution point of view, and in the above case would have reached the less efficient solution. The transshipment algorithm employs the system solution point of view. It is important to understand and to remember the points of view taken by the two solutions when comparing the results. Understanding the strengths and weaknesses of both solutions will allow the user to understand the network and to interpret the solutions.

CHAPTER 4 - METHODOLOGY AND DESCRIPTION OF THE RUNS

The purpose of this study is to determine the impacts of changes in the network cost structure on the delivered cost of coal in foreign markets. The network, consisting of the links described in Chapter two, will remain the same throughout this study. In order to measure the impacts of changes, two things are necessary. First a benchmark or common base is required against which all alternatives can be measured. Second some means of implementing the changes to the link costs must be established.

The network to be modeled is drawn as Figures 1 and 2. The two figures depict the US and foreign transportation networks, respectively, over which coal is shipped. Not present in the two figures are the vessel links which connect every US port with every foreign port. The types of links are not labeled, but they are described in detail in chapter two. In order from the origin mine to the destination utility the links traversed are: mine production, truck to the tipple, railroad to the port, transshipment at port, port user fees, vessel voyage, foreign port fees, foreign transshipment, foreign rail, and the utility consumption. The only exceptions are three barge links, one in the US (link 27-28) and two in Europe (links 67-77 and 67-78). The cost structures for all links are described in Appendix B.

As is clear from the figures, several of the links leave no options for shipments which reach their onode. The production links,

the truck links, the transshipment links on both shores, both port links, and the utility links are specific functions performed at a specific location with no practical choice in how to perform them. Every foreign port can be reached from every US port, insuring the connectivity of every origin mine and every destination utility. The US rail network and, to a lesser extent, the foreign rail network, also present some path choice to the flow of coal. This is not to say that the other links do not influence the path choice of the coal. On the contrary, coal must pass over every type of link, and every link will impact on the final delivered cost of the coal.

The US transportation network is modeled to include all the main facilities in the eastern region of the US. All the major eastern railroads are represented, and the ports include the major exporters of coal on the Atlantic Ocean, the Gulf of Mexico, and the Great Lakes. Only the Appalachian coal fields are represented in this model because they are presently the major exporters from the US. On the foreign side, all the major markets are included in Europe and Japan, along with several of the smaller markets to capture the effect of vessel size on the overall network. There has been a generic foreign port created which represents a small port of 31 foot draft to compete directly with the large ports. The small port close to the utility can then be an option to the large port with a long rail haul. The structure of this network will not change throughout this analysis. The links will remain

as they are, and connectivity will be maintained. The cost structure of several links will change, and it will be this change which will be measured to determine its impact on the network.

4.1 METHODOLOGY

The benchmark for this analysis will be the network as described with the physical limitations as they now exist. The capacities of the links will be the present physical capacities. The links modeled will be the ones presently in use, and no attempt will be made to forecast new facilities. The cost structure will be the present costs obtainable from the available data, and the origin-to-destination flows will be modeled on those of 1982. A summary of the data input for the base case is found in Appendix B.

From this benchmark the network will be changed to study the impact on the delivered cost of coal. The specific topics to be studied are changes in the port capacities in terms of the size of ship they can handle and changes in the volume of US exports. The first set of changes modeled will be the port dredging, and this will be done with all other factors held constant. The ports considered for dredging will be Baltimore, Hampton Roads, and Mobile, and they will be considered singly, in pairs, and all together. Once they are all modeled for the present 32 million

tons per year of export steam coal volume, they will be modeled again with the export volume doubled to 64 million tons per year. Each change in the network will be measured against the base case for the volume of export flow.

4.2 BASE CASE

The presently configured transportation network with the costs and O-D flows from 1982 will form the base case or benchmark against which all changes will be measured. This benchmark network is still essentially as it was in 1982, but some of the costs have changed and the O-D flows are not exactly the same. However, within the limits of current data this scenario is a good starting point for this investigation.

Only the Appalachian coal fields are modeled here, but they export nearly all the coal being shipped from the US. I have modeled the Appalachian region as six subregions ranging from Pennsylvania to Alabama. As explained in Appendix B, there are four cost groups for coal production, and I have assigned a group to each of these six subregions. This arrangement models the effects of both location and sulfur content which, with the heat content, contribute the most to the cost of coal. High sulfur coal has a sulfur content of 1.6% or higher, and low sulfur coal has 1.5% or less. The input data from Appendix B shows the exact

inputs for this scenario. The supply of coal available for export shows a greater supply from northern Appalachia than from the southern fields in keeping with present export levels. These coal subregions are constant throughout the study with the exception of the available supply which doubles when the export volume doubles.

The truck links are modeled very simply and given relatively little importance in the network. The distances are short in keeping with the network in Appalachian fields. No attempt is made to model alternate means of moving the coal to the tipple such as conveyors or extensions of the railroad lines to the mines.

The rail network is the first set of links which offer the flow of coal some path choice. Because of the high interchange costs relative to linehaul costs, the rail network in this model is based on single railroad links with no interline connections. There is some circuitry added to the links because of this no-interchange modeling in the network, but it models all the major lines to the ports in a simple and straightforward manner. Railroad costs as opposed to prices are used in this network initially. Alternate routes by different railroads are modeled to several of the ports to reflect the competition in the rail network. The northern Appalachian fields have a much stronger rail network to both the Atlantic and the Great Lakes ports than do the southern fields to the Atlantic and the Gulf ports. This reflects the current configuration of the major rail

lines in the Eastern US. One rail link (#17-27) connects the Tennessee coal fields to the several river ports which provide barge service to New Orleans. The barge traffic is not modeled in depth, and only this one representative path is provided as an option to shippers.

At each port two links are used to model the two functions of loading vessels and recapturing the investments made at the port. The first function is termed transshipment, and it includes the process of accepting the coal from the rail cars or the barges, storing the coal, and loading the coal on the vessels. In this model only the vessel loading costs are used, and no attempt is made to model different types of transshipment equipment or the inventory costs.

The port link is in place to model the investments made in the ports in the form of capital improvements, such as dredging harbors and channels, and of maintenance expenses such as silt removal. The fees are based on a 50% recovery of the investments by the government, and are inversely related to the total volume of shipping through the port. In the base scenario, only maintenance costs are considered since no dredging or other capital improvements are in place. The Great Lakes ports of Toledo and Ashtabula receive a \$2 per ton fee to reflect the tolls on the St. Lawrence Seaway. The initial port depths and the maximum vessel capacities are listed on the input menu in Appendix B. The costs to maintain the port of New Orleans are divided

between the facilities serving the rail connections and those serving the barge connections since both benefit from that maintenance, and each facility has to contribute its share to those maintenance costs.

The vessel links are the most numerous of the transportation network, and they reflect the fact that the open water can be traversed in any direction without restriction. The vessel distances are in nautical miles, and they represent the direct route between the ports concerned. Routes to Japan from Toledo and Ashtabula use the Panama Canal, but all other vessel links to Japan travel via the Cape of Good Hope because of the vessel size involved. The shipment size and draft reflect the capacity of the smaller port on the vessel link. At New Orleans the land and river facilities are modeled with the same capacities and travel distances.

The ports on the foreign end of the transportation network are treated similar to those on the US end, but the costs through those ports have been estimated and fixed. The ports chosen for the model cover a range of sizes from quite small, such as Cadiz and Casablanca, to very large, such as Rotterdam. Most of the major importing nations of the world are included such as France, Germany, Great Britian, and Japan. In addition some smaller nations are included in the model as well, such as Morocco and Finland. The generic port could be any small port in any of the nations involved. It was included to compete directly with the

very large ports, and it is always located much closer to the final user than the larger port. This allows the shipper the option of more direct shipments though at the cost of economies of scale on the vessel link. The foreign transportation network is not meant to include all possible destinations or routes, but it gives a representative sample of the options available and the routes to be taken in the export of US steam coal.

The foreign transshipment facilities are modeled in a manner similar to the foreign ports with fixed costs. No attempt is made to reflect the type of equipment present, but the link inputs on the input menu can be used to compute facility costs based on the type of operation. These links are included to model the real network, and to reflect the total costs of the network on the delivered cost of coal.

Only representative rail links have been included in the foreign transportation system, but these links have been chosen to include the choice between the generic port and the larger ports where that choice makes sense. The distances are in miles, and they are the only input which affects the link costs. Therefore the very short distances from the generic port contrast sharply with the longer rail distances from the larger ports. The difference in costs can be used to balance the larger vessel costs to the smaller ports.

The foreign utilities were chosen more for their representative locations than as individual utilities. Each utility is given the

demand for the entire country that it represents, and it is meant only as a first estimate of the cost to the importing nation. Certainly the costs will vary within the nations based on the exact locations of the using facilities. But the utilities chosen represent a cross-section of the final users of US export steam coal and the various demand rates with which each is associated.

Overall the transportation network reflects a wide range of options about routing, shipment size and origin for the export of steam coal. Limitations by importing nations such as maximum imports from any one supplier have not been built into the model, but such limits could be incorporated in the form of link capacities on the proper links. The network is flexible both in its initial configuration and in its ability to adapt to reflect changes in the cost structures of capacities of any of the links.

4.3 CHANGES FROM THE BASE CASE

There were two type of major changes from the benchmark transportation network investigated in this study. The effect of dredging of US ports was the focus of the study, and the effect of a doubling of the export volume was also studied. Dredging of the ports of Baltimore, Hampton Roads, and Mobile was investigated to determine if the ability to load larger vessels would affect the delivered cost of coal in overseas markets. The cost of dredging

was included in the recomputed port investment and maintenance costs, and it was included to determine if such an increase would offset the positive effects of larger vessels.

The study was completed in two series of eight scenarios each. The first series was performed to contrast with the benchmark network described above, and the second series contrasted with the benchmark network when it was carrying twice the O-D flows. In each series the three ports were modeled as dredged singly, in pairs, and all three together to investigate the varying impacts of those actions. Appendix D includes a listing of the resulting flow patterns and the O-D cost matrices.

When a port is dredged in this model, the costs on two links are changed. The beneficial change is on the vessel links from that port to every foreign port which can service vessels larger than the original capacity of the US port. For instance, when Baltimore is dredged to accept vessels of 110,000 tons rather than 61,000 tons, every link to a foreign port which can handle vessels over 61,000 tons is affected. Since the vessel costs decrease as the vessel size increases, this effect is positive. The negative effect of dredging is recovering the capital costs of the initial dredging and the increase in the maintenance costs at the ports. These costs are specific to the US port dredged and are spread over the total coal tonnage that moves through that dredged port. One can see, then, that dredging has a negative effect on shipments to smaller ports from the dredged port since they must

pay the port fees without the potential for cost savings on the vessel link. But there is potential for cost savings to larger ports where the gains on the vessel link may outweigh the additional cost on the port link.

The remainder of the transportation network is unchanged by the port dredging. The port of Baltimore is modeled as dredged to handle 110,000 ton vessels, Hampton Roads to handle 150,000 ton vessels, and Mobile to handle 110,000 ton vessels. The effect of this dredging on the vessel link costs is reflected in the tables of Appendix C.

4.4 CASE STUDIES

The 16 runs will now be described individually. The flow diagrams for these runs are found in Appendix D, and they show the changing flow patterns based on the dredging in the scenario. The diagrams show that the flow of coal generally shifts to a dredged port, but the origin-to-destination cost matrices, which are also in Appendix D, shows that the cost of the delivered coal is not always greatly affected.

4.4.1 BENCHMARK SCENARIO

Almost all the export coal is delivered from the northern Appalachian coal mines. Coal is cheapest to every destination from Southwest Pennsylvania, and almost all of that coal is shipped through the Great Lakes ports of Toledo and Ashtabula. Hampton Roads is the US port which handles the largest share of the US exports, shipping all the coal which is available from the Central West Virginia coal mines. All the southern coal originates in the Alabama fields and is exported through the port of Savannah.

The base case flows in the model differ a good deal from the actual flows as indicated in Table 3. Two factors that may contribute to this variance are: 1) export coal to Canada is not

PORT FLOWS

<u>PORT</u>	1980 ACTUAL FLOWS	MODEL <u>PREDICTED</u>
TOLEDO	4	8
ASHTABULA	5	7
PHILADELPHIA		
BALTIMORE	12	2
HAMPTON ROADS	19	10
SAVANNAH		5
MOBILE	5	
NEW ORLEANS	4	
	<hr/> 49	<hr/> 32

All Figures in Millions of Tons per Year

Data for Base Case (Scenario 1)

TABLE 3

included in the model but is in the actual flow data, and 2) there are uncertainties in rail costs, and 3) some companies own transshipment facilities at some locations which they use even in the face of a cost disadvantage. The first factor explains a large part of the difference in the total volumes. The two other factors together suggest that coal may be exported more cheaply through the Great Lakes ports than through the east coast ports, but that semi-fixed East Coast investments and the 3 month freeze-up may offset that advantage. Later iterations will show that coal from Pennsylvania which goes through the Great Lakes ports in the base case, goes through Baltimore when that port is dredged. It may also be that the capacity limitations at the Great Lakes ports force the same shift in flows in actual practice. The port of Hampton Roads ships most of the metallurgical coal exported from the US, but this flow is not included in the model since the model is only concerned with export steam coal. The export of the metallurgical coal may draw steam coal to Hampton Roads as well by allowing economies of scale. The southern flow from Alabama to Savannah can only be explained as a result of the railroad costs, and again in part from the lack of other limitations on other ports by other commodities. The use of only single-line rail links in this model explains why the cost of coal service to Mobile from the coal fields in northern Alabama are so high, whereas there is a direct connection to Savannah. (In fact, though, the interline

rates to Mobile are comparable to single-line rates, but were not included in this model.)

At the foreign end of the network, the generic port receives almost a third of the coal, favoring the small ports close to the destinations. Rotterdam also received a large share of the imports to Europe, and then transshipped it to all of its possible destinations. Only at the Thames Power Station was there a conflict between the large port of Rotterdam and the generic port. Here the rail link from the the generic port carried a capacity flow of 2 mtpy (which may be too low), and which leads one to believe that the small port was more efficient and less costly than the larger port to that destination.

On the vessel links, the larger US ports of Hampton Roads and Savannah shipped to Rotterdam, and Hampton Roads also shipped to Marseille. Baltimore shipped to Hamburg. The smaller US ports of Toledo and Ashtabula shipped to the smaller foreign ports of Helsinki, Cadiz, Casablanca, and the generic port. Thus the large US ports took advantage of their larger vessel capacities to handle the flows where large vessels were an advantage, but the smaller ports were able to capture the remainder of the exports and in places to compete favorably with the larger ports by shipping more directly. The flow diagrams for the base case in Appendix C reflect the link costs for all the links except the US rail and the vessel links. These link costs are tabularized to aid the reader, also in Appendix C.

4.4.2 DREDGE BALTIMORE

The flow diagram portrays a tremendous change in the flows from the benchmark when Baltimore is dredged. The flow through Baltimore jumps from 2 million tons per year (mtpy) to 17 mtpy which is its capacity. These flows are drawn from the Great Lakes ports and from Hampton Roads. Also affected are the origin regions of the coal as Central Pennsylvania's exports rise to capacity at the expense of Central West Virginia. Also of significance is the fact that the delivered cost of coal to the Central European destinations of Hanover, Apeldoorn, Frankfurt, Thames Power Station, and Lyons are decreased by \$1 - \$4 per ton. At the same time the delivered cost to the smaller importers such as Tempere, Madrid, and Marrakesh rose by about \$2 per ton for coal from Central Pennsylvania. These cost changes are quite significant since they are changes from the lowest delivered costs in the benchmark scenario. The costs to Central Europe reflect very favorably on the impact of dredging Baltimore, and the rising costs are of little significance since Baltimore does not have the transshipment capacity to handle that flow in addition to the flow to the larger ports.

The foreign transportation network showed only one change from the benchmark. Rotterdam became the foreign port for all the coal

delivered to the Thames Power Station suggesting that the advantage of the generic port over Rotterdam in the base case was a slim one, and that it was overcome by the economies of scale gained in dredging Baltimore. The other vessel routings remained essentially unchanged with the larger US ports shipping to the larger foreign ports. Ashtabula's exports are reduced to 3 mtpy because all the remaining coal from Pennsylvania is shipped through Baltimore. Savannah's trading partners are changed to smaller foreign ports to compensate for the reductions on the Great Lakes. Baltimore, as the largest US port, captured the trade with Rotterdam and Marseille, but that was the only shift from the base case.

4.4.3 DREDGE HAMPTON ROADS

The flow pattern when Hampton Roads is dredged to handle 150,000 ton vessels is exactly the same as the benchmark flows. Hampton Roads is unable to capture coal flows from any other coal producing region than Central West Virginia, and thus it has only a limited impact on the export of steam coal. The rail network from other producing regions to Hampton Roads is only slightly more expensive than the link taken from Central West Virginia to Hampton Roads, but the other areas are so much closer to the other ports that the rail costs to those other ports are low enough to

not be affected by the reduced vessel costs through Hampton Roads. The delivered cost of coal from Central West Virginia to those destinations serviced by ports of 150,000 ton capacity or more is reduced by about \$1 per ton, but this affects only Apeldoorn, Frankfurt, and the Thames Power Station, which transship in Rotterdam, and Lyons which transships in Marseille. The Thames Power Station is forced to receive coal from the generic port as well as through Rotterdam because the production of export coal from Central West Virginia is at capacity of 10 mtpy. The costs of coal from Central West Virginia to Central Europe, though reduced, are not the lowest available to those destinations. The lowest coal costs continue to be for Pennsylvania coal which is transshipped through the Great Lakes ports. The utilities in those locations would have an incentive to try to receive their coal from other US sources even though this would in reduce the system efficiency. Dredging Hampton Roads appears to not be a favorable option.

4.4.4 DREDGE MOBILE

Dredging Mobile to handle 110,000 ton vessels increases the flow of coal through that port from zero to 5 mtpy which is the capacity of the Tennessee coal producing region for export. These flows cause reductions in several other producing regions

including Central Pennsylvania and Central West Virginia. The exports handled by Mobile were handled by Hampton Roads and Toledo in the benchmark scenario. As with the dredging of Hampton Roads, the delivered cost of coal shipped through Mobile in this scenario is reduced significantly on shipments to other large ports. In this case the reduction is about \$4 per ton, but the coal starts from such a high base cost that even this large reduction is unable to match that of coal from other producing regions, such as Central Pennsylvania and Central West Virginia, shipped through smaller ports.

Dredging Mobile would appear to benefit only Mobile and not US coal exports in general.

4.4.5 DREDGE BALTIMORE and HAMPTON ROADS

The flow pattern when both Baltimore and Hampton Roads are dredged is quite different than any case previously. The positive effects of dredging Baltimore are still apparent in the costs of coal delivered to Central Europe, but the flow of coal through Baltimore to Rotterdam is cut in half when Hampton Roads is dredged as well. However the difference in Baltimore's shipments to Rotterdam is handled by the Great Lakes ports of Toledo and Ashtabula. Hampton Roads still cannot draw coal from any producing region other than Central West Virginia, but what it

does ship is cheap enough to compete with Pennsylvania coal through Baltimore. What happens is that Baltimore and Hampton Roads have too much combined capacity to handle the flow to the large foreign ports, and they do not compete well with the Lakes ports on shipments to smaller foreign ports.

4.4.6 DREDGE HAMPTON ROADS and MOBILE

The ports of Hampton Roads and Mobile draw coal from different producing regions, but they do not ship enough coal to supply the all of the utilities' demands which could flow through the large foreign ports. Because each port only draws from one coal producing region, its impact on the export system is limited. Coal through both ports goes to the larger foreign ports of Rotterdam and Marseille and in each case the dredging reduces the delivered cost of that coal significantly. But in neither case is the cost of the delivered coal as low as that from other US sources.

The Great Lakes ports, especially Ashtabula, continue to be economical ports of departure for US coal to foreign destinations serviced by small ports. Ashtabula is operated at capacity in this scenario, and the coal is shipped from there to Helsinki, Casablanca, and the generic port. Toledo ships to the generic port as well. The Great Lakes ports are able to capture the

traffic to the smaller foreign ports where the benefits of large vessels cannot be realized.

4.4.7 DREDGE BALTIMORE and MOBILE

In this scenario the ability of the port of Baltimore, when dredged, to draw coal from more than one producing region is of major importance. Mobile handles no export coal even though it is dredged to handle 110,000 ton vessels and though that ability reduces the delivered cost by up to \$4 per ton on coal delivered to Europe through Mobile. Baltimore is able to draw enough coal from the two Pennsylvania coal regions to handle the shipments to the large foreign ports. The combination of reduced vessel link costs and low rail costs due to proximity to the coal fields gives Baltimore this ability. Hampton Roads shipped some coal to Rotterdam which was surprising based on the results of run 2, and Ashtabula continued to function at capacity to the small foreign ports. Savannah also shipped to smaller ports which was a change from previous scenarios when it had shipped to larger ports such as Rotterdam.

Coal through Baltimore to Central Europe continued to be cheaper when the port was dredged. Coal to Hanover was reduced by \$1 per ton, to Apeldoorn by \$3, to Frankfurt by \$3, to the Thames Power Station by \$1, and to Lyons by \$3 per ton. Coal shipped through the smaller foreign ports from Baltimore rose in cost by about \$2

per ton. But these shipments were more expensive than the shipments to the larger ports, and Baltimore's transshipment capacity could not handle that combined amount of flow. Therefore only the cost decreases to Europe would be realized, and the dredging of Baltimore appears to be beneficial.

4.4.8 DREDGE BALTIMORE, HAMPTON ROADS and MOBILE

This scenario results in exactly the same flow pattern as scenario 5 when only Baltimore and Hampton Roads were dredged. The flow to Rotterdam marks the only advantageous flow through Hampton Roads over the flow to a similar destination through Baltimore. Baltimore handles the large vessel flows to Marseille and Hamburg, and the remainder of the flow to Rotterdam. Ashtabula again handles flow at capacity to the port of Helsinki and the generic port. Savannah handles the remainder of the exports, serving Casablanca, Cadiz, and the generic port.

Costs through Baltimore to Central Europe again are decreased as the port is dredged. In every case of this dredging, the costs per ton to Hanover are reduced from \$56 to \$55, to Apeldoorn from \$60 to \$57, to Frankfurt from \$54 to \$51, to Thames Power Station from \$52 to \$ 51, and to Lyons from \$65 to \$62. Regardless of what is happening to any other port, these cost reductions are realized for coal shipped through Baltimore, and that port draws

additional flows to itself. Baltimore cannot handle all of the exports to destinations where the costs are reduced because of capacity restraints. However in this scenario, Baltimore does not export at its capacity because of competition from Hampton Roads. Again Mobile does not handle any export coal even though it is dredged, indicating that it should not be dredged if any other port is dredged.

4.4.9 BASE CASE AT 64 MTPY

This scenario is the base case for the second set of eight runs where the export volume is doubled to 64 mtpy. The network structure is returned to its original state, and all the US ports are modeled at their present capacities. There are two major differences between this case and the benchmark scenario in addition to the total volume. First the foreign network is nearly at capacity once the foreign port links are reached. Second the production costs are much higher when the output is doubled than in the original case.

When the export quantity was doubled to 64 mtpy, the utility demands were not uniformly doubled to obtain larger demands across the range of destinations. Instead the coal destination forecast for 1985 from the Appalachian Regional Commission report (11) was used. Under that forecast, demand in Japan would jump from 3 mtpy to 11 mtpy, demand in France went from 5 to 6 mtpy, in Germany from 6 to 16 mtpy, and in Great Britian from 4 to 16 mtpy. In recent years the actual demands have lagged the forecasted demands, and these figures may be a little different than actual flows. However they do reflect a geographical distribution which is a reasonable approximation.

The first thing that is obvious from the flow diagrams is that the larger quantity of flow is spread across more producing regions and shipped through more ports than were the benchmark flows.

This impression is valid; however, the flow quantities still show that 75 % of the export coal comes from the northern Appalachian coal fields. The flow pattern in the US shows a broadened base from which the flows are generated, incorporating five of the six producing areas.

Hampton Roads still commands the largest throughput of any US port by handling all the flow produced by the Central West Virginia coal mines. That flow of 20 mtpy is almost a third of the total exports. The Great Lakes ports of Ashtabula and Toledo still supply the majority of the coal to small foreign ports.

Philadelphia has gained a small share of the export trade under this scenario of doubled volumes, which it ships to Hamburg. In the south Mobile handles the entire 10 mtpy available from the Tennessee coal fields and ships it to the Japanese port of Nagoya. Savannah picks up the remainder of the flow which comes from Alabama, and it ships to Rotterdam.

On the foreign side, nearly every link is in use to fill the utility demands. However some of the previous flow rules still remain constant. Rotterdam received 22 mtpy which is over a third of the total volume exported from the US. The generic port is filled to capacity reflecting the continuing ability of the small port to compete close to the utility to compete with the large port at a greater distance from the utility. One major change is observed in the shipments to Rome. In all previous scenarios the path was through the generic port, but in this scenario the port

of Tarranto was used for the entire flow quantity. This quantity was increased from 3 mtpy to 5 mtpy which may indicate that the change was due more to the generic port being at capacity than a change in the economics of Tarranto. This may indicate that the generic port in the model should have been given more capacity because there are many "generic" ports.

The ports connected by vessel links continue to display the larger port-to-larger port and smaller port-to-smaller port pattern. The flow from Hampton Roads goes mainly to Rotterdam (15 mtpy) with the rest to Marseille. Savannah fills the remaining demand through Rotterdam. Baltimore ships to Marseille and Tarranto, and Mobile ships to Nagoya in 61,000 ton vessels via the Cape of Good Hope. Although the costs have increased sharply across the board, the flow patterns have not changed much at all. The biggest difference is the broadening of the flows to accommodate the increased demands, and this broadening occurred at both ends of the transportation network. This scenario will serve as the benchmark for the second series of runs.

4.4.10 DREDGE BALTIMORE

When the port of Baltimore is dredged, the flow pattern is changed a great deal from the benchmark to reflect the low cost possibilities of that port. Although the same five coal producing

regions still export coal, there is a shift from Tennessee and Central West Virginia to Central Pennsylvania. This is due to the ability of Baltimore to draw coal from the two Pennsylvania producing regions and to influence the balance of coal exports from the US. Baltimore captures the trade to Nagoya, Japan and to Marseille, as well as the majority of the shipments to Rotterdam. The exports through Hampton Roads are diminished and split between Hamburg, Rotterdam, and Tarranto. Ashtabula continues to ship at capacity to the generic port, and Savannah and Mobile split the traffic to the remaining small foreign ports.

At the foreign end the flow pattern does not change from the base case except in the origin ports for the vessel links. Once the coal reaches the foreign ports the flow pattern and the quantities are exactly the same as the base case.

Dredging Baltimore has a strong downward pull on the delivered costs to Central Europe, but, unlike the lower volume scenario, it does not cause the costs to rise to the smaller ports on coal from Central Pennsylvania. The cost reductions affect both Pennsylvania producing regions, and they range from \$1 to \$4 per ton. Because the cost reductions affect two producing regions and are measured from the lowest costs in the base case, they have a strong impact on the flow pattern of the delivered coal. Again it is the ability of Baltimore to draw coal from two producing regions which allows that port to strongly influence both the flow pattern and the delivered cost of coal.

4.4.11 DREDGE HAMPTON ROADS

The flow pattern shows that the increased volume works in the favor of Hampton Roads when that port is dredged. For the first time the port is able to draw coal from a Pennsylvania coal field and to significantly change the flow pattern. From the northern coal fields, the coal flows either to Hampton Roads or to a Great Lakes port. This is a significant change in direction, and it shows that Hampton Roads' importance will increase as the volume of US export coal increases. In the south the flow pattern reverted back to that of the base case with the Tennessee coal fields again producing at capacity.

Hampton Roads ships coal only to the major ports of Rotterdam, Nagoya, and Tarranto. Savannah and Mobile ship to the other major ports of Hamburg and Marseille, while the Great Lakes ports continue to handle the flow to all the smaller foreign ports. The flow pattern once the coal reaches the foreign ports was again exactly the same as in the base case.

The delivered cost of coal to Central Europe was only slightly lower with the dredging of Hampton Roads than in the benchmark. However dredging the port did not affect the lowest deliverable costs of US coal. The lowest costs could be obtained by shipping Southwest Pennsylvania coal through the port of

Ashtabula. Almost certainly the short rail haul is the key to that connection since Ashtabula is a very small port and cannot compete on the vessel link.

4.4.12 DREDGE MOBILE

There is no difference in the flow pattern in the US when Mobile is dredged in this scenario and when it is not dredged in the benchmark scenario. Not only are all the same links used, but the quantities are within 8 mtpy of being exactly the same on the US network. With the dredging, the flows from Mobile are split between Rotterdam, Tarranto and Nagoya rather than all going to Nagoya. This causes the shifting of flows on several vessel links and results in an increase of flow from Toledo at the expense of flows from Baltimore. This is because dredging Mobile changes the total costs of delivered coal through that port relative to the delivered costs through other US ports, and it forces the US ports to shift the foreign ports to which they ship accordingly. Hampton Roads continues to provide the majority of the coal to Rotterdam, and it ships to Marseille as well. Baltimore ships only to Nagoya while Toledo ships to Cadiz and the generic port. Ashtabula ships to the small ports of Helsinki and Casablanca. Again the foreign flow pattern is unchanged from the base case.

Dredging Mobile causes a significant decrease in cost for coal shipped from Alabama to Central Europe, averaging \$4 per ton. However, these costs started from a base of more than \$10 per ton greater than the costs from Pennsylvania, and the \$4 does not change that relationship very much. Also Mobile failed to attract any more coal in this scenario than it did in the base case. The impact of the dredging is a local impact and not very important to the total export system.

4.4.13 DREDGE BALTIMORE and HAMPTON ROADS

Dredging Baltimore and Hampton Roads together appears to be a compromise from dredging either one of them separately. Baltimore is no longer able to replace flows from Central West Virginia through Hampton Roads with flows from Central Pennsylvania through Baltimore. And Hampton Roads loses its ability to draw flow from Southwest Pennsylvania. From the base case, the spread out flow among the northern ports is more concentrated, predominantly at the two dredged ports. Toledo and Ashtabula continue to ship to the generic port and the other small ports, but almost two thirds of all US exports flow through Baltimore and Hampton Roads, split almost in half. Baltimore ships to Rotterdam and Marseille, and Hampton Roads ships to Nagoya, Tarranto and Rotterdam. Savannah and Mobile are left with only small foreign ports to

which to ship.

Delivered costs to Central Europe fall for all three northern producing areas by the same amounts as when the respective US ports were dredged separately. Since these areas have the lowest costs initially, these reductions increase the differences.

4.4.14 DREDGE HAMPTON ROADS and MOBILE

The flow diagram for this scenario is exactly the same as for dredging Mobile by itself and as for the benchmark. In fact the flows within the US transportation network are exactly the same as the base case. With the foreign network also being exactly the same as in the base case, only the routing of some of the vessel links are different. The flow from Hampton Roads to Rotterdam is reduced from 15 mtpy to 5 mtpy, and the remaining 15 mtpy of Hampton Roads' total export volume of 20 mtpy is split between Nagoya (10 mtpy) and Tarranto. Mobile ships to Marseille and Rotterdam. Savannah and Baltimore also ship to Rotterdam. Toledo and Ashtabula split the small ports as they did in the base case, and Philadelphia regains the trade with Hamburg.

On the delivered costs, again improvements are made on shipments through the dredged ports. Of the two ports, costs are reduced more on shipments through Mobile than through Hampton Roads, and coal from Tennessee can now be delivered cheaper than coal from

Central West Virginia to Central Europe. But neither can be delivered as cheaply as can coal from Southwest Pennsylvania through the Great Lakes ports in this scenario. Overall there is very little impact from the dredging in this scenario over the base case, and it indicates no advantage to dredging Mobile if Hampton Roads is dredged.

4.4.15 DREDGE BALTIMORE and MOBILE

Baltimore ships almost half of all coal exported from the US in this scenario, and the dredging of Mobile allows it to ship all the available coal from the Tennessee coal mines which it was not able to do when Baltimore was dredged by itself. This improvement at Mobile comes mainly at the expense of flow through Hampton Roads. Baltimore captures the entire 22 mtpy shipped to Rotterdam as well as shipments to Marseille and Nagoya. Mobile ships to Tarranto and Nagoya. Hampton Roads retains the remaining shipments to Marseille and Hamburg. Savannah, Ashtabula, and Toledo again handle the trade with the small foreign ports. Again the transportation network at the foreign end remains as it was in the base case.

Southwest Pennsylvania coal shipped through Baltimore continues to have the best delivered cost available in Central Europe. This cost is reduced by the same \$1 to \$3 per ton from the base case

when Baltimore is dredged. Mobile continues to reduce the cost of delivered coal more than the other dredged ports, but even with these reductions the cost of Tennessee coal in Central Europe is \$10 to \$12 per ton higher than coal from Pennsylvania.

4.4.16 DREDGE BALTIMORE, HAMPTON ROADS and MOBILE

This flow pattern and the previous one look exactly alike, but the dredging of Hampton Roads has changed the flow quantities. Baltimore is reduced from one half of US exports to one third, and Hampton Roads captures all of the change. At the same time the amount of coal from Central Pennsylvania was reduced by 9 mtpy, and that from Central West Virginia is increased by the same amount. Baltimore exported all of its coal to Rotterdam, and Mobile supplied the remainder there. Mobile also supplied coal to Hamburg and Marseille. Hampton Roads shipped to Tarranto and Nagoya. Helsinki, Cadiz, Casablanca, and the generic port continued to be supplied by Toledo, Ashtabula and Savannah.

The effects of dredging all three ports were reflected in the delivered costs of the coal. It is interesting to note that the influence of the ports extends only to one or two coal producing regions, and that the influence of the port on delivered costs is constant regardless of the combination of ports being dredged. In keeping with that reasoning, coal shipped from Southwest

Pennsylvania through the port of Baltimore when it is dredged continues to be the least expensive option to Central Europe.

4.5 SUMMARY of the RUNS

Two results appear to be consistent over the 16 runs to this point. First dredging any port has a positive impact only on the coal which moves through that port to foreign ports which are large enough to accomodate the larger vessels which the dredged port can handle. Second, throughout all the runs, Ashtabula consistently shipped at capacity to the smaller foreign ports which could not benefit from the dredging of US ports. The first result is one I do not find surprising, but the second one I had not expected.

Ashtabula, Ohio is located quite close to the Southwest Pennsylvania coal fields and within only a slightly longer distance of the Central Pennsylvania and Central West Virginia coal mines. The ships which call on the port are limited to 26,000 tons by the St. Lawrence Seaway, and the only port fees modeled were the tolls for using the Seaway . Ashtabula, then, appears to be well positioned to continue to serve the smaller foreign ports. The only limitation on Ashtabula in this model is its transshipment capacity which is only 7 mtpy. In actuality, Ashtabula is limited by the freezing of the Seaway to service for

only 9 months of the year. Within the scope of that limitation, it would be a good port for which to examine improved transshipment facilities without increasing the vessel size.

On the matter of dredging among the ports modeled, only Baltimore showed a clear advantage over the whole range of options studied. Baltimore was the only port which could change the lowest delivered cost of the coal to Europe. Baltimore was also the only port which could draw coal through its facilities from more than one coal producing region over the whole range of options investigated. This suggests that there are more conditions under which dredging Baltimore will have a major impact than for any other option. For that reason Baltimore should be the first port considered for dredging.

Several other factors are stable during the 16 runs. The foreign flows took on one pattern during the base case and another pattern when the export volume was doubled. However in each case the flow pattern was remarkably stable over the series of eight runs. This consistency of the foreign network held even when larger economies of scale were available as the US ports were dredged. There was only one change in the foreign flow pattern was that the flow to the Thames Power Station through the generic port vanished as soon as the first US port was dredged, and that flow only went back to the generic port when any US port was dredged and did not have the capacity to fill the demand through Rotterdam (run 3). Aside from this one change from a small port to a larger port, the dredging

did not change the type of markets served by the US ports. Larger US ports consistently shipped to larger foreign ports, and the smaller US ports consistently shipped to the smaller foreign ports. The dredging seemed only to change the relative advantage of the larger US ports in their competition with each other.

Another strong consistency was the US coal producing areas which were able to export coal. The two Pennsylvania regions and Central West Virginia always exported the majority of the coal, with Southwest Pennsylvania being the most consistent. Alabama was able to ship coal consistently through Savannah during the high volume export series, and the Tennessee coal fields were usually able to export coal through Mobile when that port was dredged.

All of these consistencies are interesting, but the impact of dredging Baltimore and the ability of Ashtabula to consistently ship coal to small foreign ports are the most important results obtained. Baltimore makes its impact because it is able to attract coal from two producing regions, and therefore it can gain the economies of scale provided by the larger vessels. This ability, and the fact that it was shipping low cost Pennsylvania coal, allow Baltimore to reduce the cost of coal delivered in Europe. Foreign ports which cannot benefit from the dredging of US ports took advantage of Ashtabula's proximity to the Pennsylvania coal fields which kept the rail costs low. This allowed Ashtabula to dominate these routes under any dredging

scenario.

4.6 SENSITIVITY ANALYSIS

Two additional scenarios were executed to test the ability of Baltimore to change the flow pattern and lower the delivered cost of US coal under different circumstances. The first additional condition was a surcharge on the cost of high sulfur coal because of the potential cost of scrubbing equipment which may be required if that coal is used. The second condition was to use railroad prices rather than costs for the US railroads and to alter the vessel link costs to reflect a diminishing return to scale from the dredging of the ports. The link costs used in this sensitivity analysis are found in Appendix C. The results of these additional scenarios are outlined below.

4.6.1 SULFUR SURCHARGE

When the sulfur surcharge of \$4 per ton was added to the mine production costs of high sulfur coal, the flow pattern with no ports dredged was very similar to the base case in the original study. Hampton Roads carried the capacity flow of 10 mtpy from the Central West Virginia coal mines and shipped it to Marseille

and Rotterdam. Toledo also exported 10 mtpy from Central Pennsylvania, and it shipped it all to the generic port. Ashtabula exported at its capacity of 7 mtpy to Helsinki, Hamburg, Marrakesh and Cadiz. In addition Philadelphia exported 3 mtpy as the available coal from all three northern producing regions was exported. It is interesting to note that the Great Lakes ports continue to carry a heavy share of the exports. It is also interesting that, with the increased production costs, the small foreign ports, including the generic port, handle a large share of the coal.

When the same production costs were used in a scenario with the port of Baltimore dredged, the flow pattern was changed drastically. Baltimore increased its exports from zero to 13 mtpy at the expense of Toledo and Philadelphia whose flows both went to zero. Ashtabula still shipped at its capacity. The dredging of Baltimore increased the shipments to Rotterdam from 9 mtpy to 12 mtpy as the economies of scale overcame the advantages of the generic port. The cost of coal in Europe was initially higher in these scenarios than in the base case, as is to be expected. But the effect of dredging Baltimore is to lower the cost by \$2 to Hamburg, by \$4 to Apeldoorn, Frankfurt and Lyons, and by \$2 to the Thames Power Station. The cost of the coal from Baltimore through the smaller ports rose by about \$2 per ton as in the low volume case when Baltimore was dredged in the original series of runs.

The result of this experiment is to reinforce the conclusion that

dredging Baltimore has an important impact on the flow pattern of US export coal and on its delivered cost. Also of interest is the fact that Ashtabula continued to ship at capacity in both cases presented here. When Baltimore was dredged, Ashtabula's source became Central Pennsylvania rather than Southwest Pennsylvania as in previous scenarios, but it was still competitive enough to retain the trade with the smaller foreign ports.

4.6.2 RAILROAD PRICES and NEW VESSEL RATES

In this scenario with a new cost structure for the vessel links and railroad prices rather than costs, the immediate impact is to favor the port of Baltimore over the port of Hampton Roads. In the original base case Baltimore handled very little export coal, but even without dredging it is exporting almost one third of the total volume in this scenario. The Great Lakes ports together are shipping almost one half of the total exports which is consistent with previous results. The flow diagram shows that the coal exported through Toledo originates in Central West Virginia which had previously shipped through Hampton Roads. These changes in the flow are because of the change from rail costs to rail prices. This change caused coal from Central West Virginia to be less expensive to Toledo than coal from Central Pennsylvania. The relative rail rates of coal from Pennsylvania

and West Virginia to Baltimore and Hampton Roads remain unchanged. Baltimore ships all the coal originating in Southwest Pennsylvania which is consistent, but Ashtabula ships the coal exported from Central Pennsylvania which is a change from the original cases. With Toledo receiving its export coal from Central West Virginia and Baltimore receiving its coal from Southwest Pennsylvania, Ashtabula must receive its coal from Central Pennsylvania even though the rail price from Southwest Pennsylvania to Ashtabula is less than the price from Central Pennsylvania to Ashtabula. The relative advantage is with Baltimore because of the economies of scale due to dredging. In the south, Savannah ships the total exports available from Alabama which is also consistent with previous runs of the model. The foreign imports continue to go mainly to Rotterdam and to the generic port, with each of them handling almost one third of the total volume. It is interesting to note that neither of those ports is at capacity in this scenario.

When the port of Baltimore is dredged under these conditions, it is still able to influence the flow pattern and the delivered cost of US coal exports. Although Baltimore handled a higher than previous export volume without dredging, it is still able to attract more coal when it is dredged. This new coal comes from Central West Virginia, and it marks the first time that Baltimore has shipped coal from that region. Hampton Roads ships no coal in this scenario, and the volume through Toledo is also diminished.

Ashtabula's export volume remains unchanged at its capacity of 7 mtpy as it continues to be the US port of choice for the small foreign ports. At the foreign end, Rotterdam receives coal at its capacity from Baltimore, and it captures the entire volume of shipments to the Thames Power Station. This increase in the flow to Rotterdam is at the expense of the generic port, and it reflects that the economies of scale gained through dredging Baltimore are sufficient to overcome the short rail haul advantage of the smaller port.

The delivered cost of the coal is also affected on shipments through Baltimore, but not as much as with the original vessel cost structure. By allowing for the diminishing returns to scale of dredging, the cost fell by only \$1 to \$2 per ton when delivered to Central Europe. Coupled with the fact that the base cost is higher in this scenario, the impact of the cost reductions is lessened.

4.6.3 CONCLUSIONS

The sensitivity analysis done here bears out the ideas that dredging Baltimore has positive effects on the export of US steam coal and that the port of Ashtabula should be examined to determine the impact of increasing its transshipment capacity. Both of these ports continued to exert a strong influence on the

flow pattern and, in Baltimore's case, on the delivered cost of US coal in foreign markets.

AD-A146 013

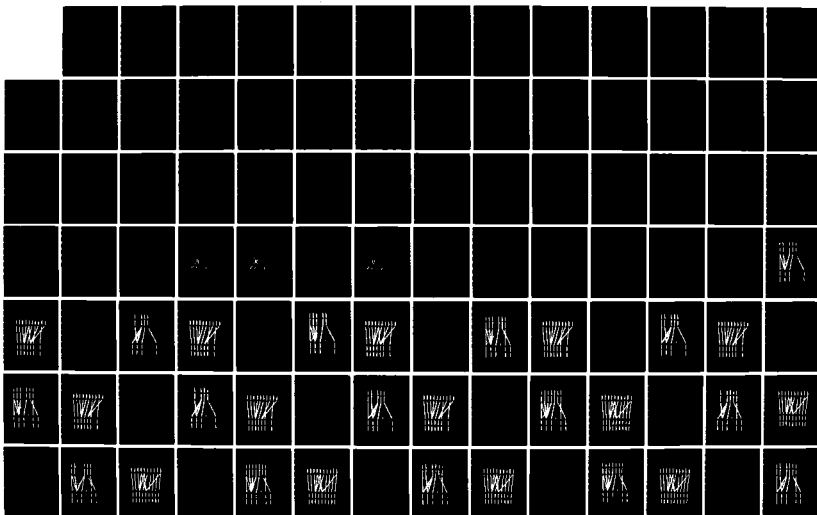
ANALYSIS OF THE TRANSPORTATION NETWORK FOR THE EXPORT
OF US STEAM COAL(U) MASSACHUSETTS INST OF TECH
CAMBRIDGE S R LINDBERG SEP 84

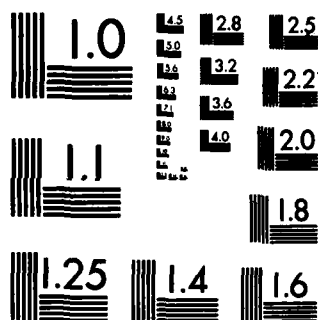
2/3

UNCLASSIFIED

F/G 5/3

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

5. COMPUTER ISSUES

The transportation network model was written using the FORTRAN 77 programming language and for the APPLE II computer. The screen handler input routine was written by John Uppgren for the same computer but using the PASCAL programming language. The use of these tools suggests several issues of which the user should beware. The different languages for the input and network routines will not hamper the usability of the product, but it does have implications for the maintenance and expansion of the program. The speed of execution of both the screen handler input routine and the network algorithm is slow, and the user must be aware of how he impacts on that speed. Finally the program takes the APPLE computer to the limit of its usable memory, and the user must know how large a network the computer will handle.

5.1 PROGRAM MAINTENANCE

The network algorithm and the input routine were written in different languages. The output of the input routine has been written to the diskette in such a manner that the FORTRAN program reads it as if it had written it. For the user, then, there will be no problem because of the different languages. In fact the user never has to know that they are different.

Every program, however, requires maintenance to keep it current. This program also has areas in which expansion is possible which are outlined below in section 5.4. It will take knowledge of both languages to perform these tasks. Both of the programs are written in a structured format in an attempt to improve understanding of the algorithms and to isolate problem areas. This structure should help as the programs are maintained, and it should allow expansion by replacing subroutines rather than by extensive rewriting of the algorithms.

5.2 PROGRAM SPEED

Speed of execution is a concern to every user, and this program is not fast. The input routine for the network tested in this thesis required 17 screens to input the 191 links. The parameter data and the origin-to-destination flows also had to be input. Unfortunately the input routine was not written to maintain the data on those screens and to use it again if there were no changes between runs. This means that all 17 screens have to be reviewed for each run, even if there are only minor changes. This process is time consuming.

The network algorithm is also quite slow, but the user has some control over that portion of the program. Two of the inputs on the parameter menu are the number of iterations and the

convergence criterion. By setting these values appropriately, the user can limit the run time of the network algorithm. For the network used here, each iteration took about 7-8 minutes to run. The user should remember that the program always executes one more iteration than is set on the parameter menu. The default setting for the iterations is 2 which caused the algorithm to execute 3 iterations and to run for about 22 minutes for this network.

The user also sets the convergence criterion which can cause the algorithm to stop before the programmed number of iterations is performed. If the user is satisfied with a higher level of variability in the network flows from equilibrium, he can increase the magnitude of the convergence criterion from its default of .02. This will cause the algorithm to stop before the number of iterations which was set if the proper level of convergence is met or surpassed.

Of course if the user changes the size of the network he will also change the running time of the algorithm. The execution time appears to be slightly more than a linear proportion to the size of the network. An earlier network of 47 nodes and 100 links took about 3 minutes per iteration to run. Therefore, when the size of the network was doubled, the run time was increased by a factor of about 2.5.

5.3 COMPUTER MEMORY

The biggest problem with this program that the user will encounter is the limited internal memory of the APPLE computer. A look at the source code for the network algorithm will reveal that the array lengths have been sized differently according to their purpose. Those used for input data are longer than those used in the manipulation of the algorithm. This is possible since the sorting routine packs the necessary data into the leading portions of the arrays. What this means to the user is that the available memory space has been maximized. The largest network which can currently be run with this program is one where the number of links plus the number of supply nodes (origins) plus the number of demand nodes (destinations) is no greater than 215. In the present network that number is 207.

There are a couple of ways in which the user memory can be better utilized. At the present time the limitation on the network is the length of the manipulation arrays which count the number of network links. These arrays can be lengthened and the input arrays shortened if the user uses the numbering system of numbering his supply nodes starting at 1 and his demand nodes immediately following his supply nodes. This numbering system will aid the formation of the initial solution of the transshipment algorithm and increase the amount of usable memory

space. This will add about 20 links to the maximum network size, and that will approach the very limits of the internal memory of the computer.

A better solution is to transfer the entire program to a computer with more internal memory. This will have the immediate effect of allowing examination of larger networks, and it will probably improve the execution speed as well.

5.4 PROGRAM EXPANSION

There are several improvements planned for this program which will give the user more options in its use. One of the first will be to update the screen handler input routine so that the user only has to make those changes he desires and the remainder of the network is entered as it was. This will allow several runs to be done on a single network in a short time and will allow an in-depth study of individual link changes. The key to this improvement is that it will increase the speed of the input routine and make the program more usable.

It was mentioned previously that the input routine calculates the price of each link as well as the cost. A future improvement will allow the user to choose which value to pass to the network algorithm. This will give the user flexibility in the use of the input, and it can be used to measure the impact of several

different pricing options based on the same cost structure.

The final update of the algorithm envisioned at this time will allow the user to choose whether he wants to execute both the flow-dependent and the transshipment algorithms or just one of the two. The user may have a need for only one of these capabilities, and it is then a waste of his time to have to run both algorithms to get the results he desires. Once the user understands the purposes and the limitations of each algorithm, there is no reason that they cannot be separated in execution.

BIBLIOGRAPHY

1. Bradley, Gordon H., Brown, Gerald G., and Graves, Glenn W., "Design and Implementation of Large Scale Primal Transshipment Algorithms," Management Science, The Institute of Management Sciences, 1977.
2. Horowitz, Elias and Sahni, Sartaj, Fundamentals of Computer Algorithms, Computer Science Press, Inc., Potomac, Maryland, 20854, 1978.
3. Kolstad, Charles D., Abbey, David S. and Bivins, Robert L., "Modeling International Steam Coal Trade," Los Alamos, Nevada, January 1983.
4. Lipfert, F. William, "Transportation of US Export Coal through Eastern Ports," Thayer School of Engineering, Dartmouth College, Contract DE-AC01-81FE-20036, June 1982.
5. Malone, Frank, "For Coal, a Decade of Decline," Railway Age (April 1983), pp. 38-40.
6. Sheffi, Y., Urban Transportation Networks, Equilibrium Analysis with Mathematical Programming Methods, Prentice Hall, Englewood Cliffs, New Jersey, 1985.
7. "Coal Export Port Study," Dravo Corporation, Appalachian Regional Commission, January 1982.
8. House of Representatives, "Report 97-454 (Part 1), Port Development," Congressional Printing Office, March 9, 1982.
9. "Market Guide for Steam Coal Exports from Appalachia," ICF Incorporated and JP Coal Associates, Appalachian Regional Commission, March 1982.

10. "Market Guide for Steam Coal Exports from Appalachia, Appendix B: Detailed Country Data and Maps," by ICF Incorporated and JP Coal Associates, Appalachian Regional Commission, March 1982.
11. "Potential Role of Appalachian Producers in the Steam Coal Export Market, Task #1: International Steam Coal Trade Analysis," ICF Incorporated, Appalachian Regional Commission, November 1981.
12. "Railroad Maps - the Midwest," Mac Publishing, Inc., Colorado Springs, Colorado, 80933, 1980.
13. "Railroad Maps - the North," Mac Publishing, Inc., Colorado Springs, Colorado, 80933, 1980.
14. "Railroad Maps - the South," Mac Publishing, Inc., Colorado Springs, Colorado, 80933, 1980.
15. "Steam Coal Import Requirements in Turkey, Egypt, Finland, and Israel," ICF Incorporated, Appalachian Regional Commission, December, 1982.
16. "Transporting Export Coal from Appalachia," by Booz-Allen and Hamilton, Inc., Appalachian Regional Commission, November, 1982.
17. US Department of Transportation, Federal Railroad Administration, "Railroad Freight Traffic Flows 1990," December 1980.

APPENDIX A PROGRAM OPERATING INSTRUCTIONS

The operation of the coal network model is straightforward. Having the proper operating environment in the form of the proper hardware and peripherals is the most difficult task. Once this is accomplished, the operations are almost self-explanatory.

This coal network program is written in the Apple Fortran programming language with a menu-driven input routine which is written in the Apple Pascal programming language. To accomodate both languages, the Apple II or II Plus computer must be equipped with the UCSD Pascal operating system version II.1 and the Apple language card. The UCSD Pascal operating system converts the program into executable P-code and then uses the 6502 Assembler, which is a part of that operating system, to convert the assembly-language routine into machine-language code. The Apple Fortran language used is the American National Standard (ANSI) Subset of Fortran 77, and it is compatible with the UCSD Pascal operating system. The Apple computer must be configured to include upgraded random access memory to 64k, and it must allow the terminal to display 80 columns on the screen. A Videx Videoterm Board was used in the program development, and it is recommended for the 80 column display. The program configuration requires that at least two diskette drives be available with the Apple computer, and a printer is required for the network output.

Once the user has the proper hardware, the program is quite simple

to run. There are three program diskettes which include all the operating instructions and files to retain the data. These three diskettes are labeled COAL1:, COAL2:, and COAL3:. The user should then boot up his system and insert COAL1: AND COAL2: into the diskette drives. When the operating system command line appears, the user should turn on the internal swapping by depressing the S-key and responding Y (yes) to the prompt which follows. This swapping is required to execute the network routine.

Execution is begun from the command line by typing X. The prompt "Execute what program ?" will appear, and the user should respond by typing "COAL1:MAIN1". A carriage return will start the program. From this point the coal network program will guide the user through the required steps of network formation and data input and through the execution of the network algorithms. When the user finishes his input on the second vessel data screen and desires to enter additional vessel link data, he will be prompted to replace the COAL2: diskette with the COAL3: diskette. The user must insure that the COAL1: diskette remains in the disk drive throughout the entire operation, especially when this replacement of COAL2: with COAL3: takes place. COAL1: contains the program instructions, and it also collects the input data to run the network algorithms.

After the user has entered all his data on the input menus, the network routine will be executed. The user will immediately be prompted about which algorithms he wishes to execute. The

flow-dependent algorithm will execute properly for any set of input using the guidelines of Appendix B. However the transshipment algorithm will only execute properly if fixed costs and capacities are entered for every link. The user is referred to Chapter 3 for a discussion of the two algorithms. If the user chooses the transshipment algorithm alone, he will be prompted again to insure that he has entered fixed costs for every link.

The network used in this thesis uses nearly all the internal memory of the Apple computer. This issue is discussed in detail in Chapter 5, and a list of the base case input menus is attached to Appendix B. The user must insure that the network he models will meet the required network size limitations, or the results obtained will be faulty.

When the user is first entering his data to the input menus, he should have a network diagram with him to insure the connectivity of the network. Without this connectivity, the algorithms will not produce an intelligent solution. Appendix B provides a guide to the input menus and the data required by each to properly run the program.

The menu-driven input routine makes this network program very simple to execute. If the user replies carefully to the prompts, he will have no trouble with any portion of the program.

APPENDIX B LINK COST COEFFICIENTS - FLOW-DEPENDENT ALGORITHM

The key to the flow-dependent algorithm is the fact that the link costs change when the flows on the links change. In order to correctly identify the changes in those link costs, it is necessary to write an equation which accurately reflects the behavior of the link costs over a range of link flows. The basic link cost equation used in this model is $COST = C_0 + C_1 * (FLOW) ** C_2$. Here C_0 represents the fixed cost on the link, and $C_1 * (FLOW) ** C_2$ is the variable cost. The fixed cost is the cost per ton of using the link whether there is any flow on it or not.

The variable cost is that portion of the cost which is directly attributable to the flow on the link. Usually as the flow increases, the cost increases due to congestion and delay on the link. This is called diseconomy of scale. In this case the exponent, C_2 , will be positive. However some links become more efficient as the flow increases, and the marginal costs decrease. These links exhibit economies of scale, and they are identified by negative exponents, ($C_2 < 0$). The magnitudes of the multiplier, C_1 , and the exponent, C_2 , reflect the steepness of the marginal cost curve on the link.

For each type of link, the cost coefficients have been computed differently to reflect those different components which influence the costs. The cost elements have been identified from several sources for the links at the US end of the transportation network. Data on the foreign transportation links is very scarce, and for many links

estimates have been used where no data was available.

Examples of the menus onto which the user enters his input data are contained in this Appendix. The data entered on these menus is the data which was used in this thesis to establish the benchmark or base case flow pattern from which all changes were measured. There are more input variables listed on these menus than are required by the model as it is currently configured. These additional variables have been included on the input menus for possible future extensions of this model. Unless an input variable is specifically referenced in this Appendix, the variable is not used in the calculation of the link cost parameters.

The production costs for US coal vary with the region and with the heat and sulfur contents. Since bituminous coal is the predominant export steam coal, the heat content factor has been included in the regional figures for production costs. The Appalachian coal region has been divided into two sections, north and south, and each section has been divided into high sulfur and low sulfur. When the user enters the region and the sulfur content on the mine production menu, (see menu 1) the proper cost coefficients are registered for the production link as follows:

	north-low	north-high	south-low	south-high
C0	30.00	29.00	32.00	23.00
C1	.086	.0275	.50	.35

● 2019年1月1日起，企业发生的符合条件的广告费和业务宣传费支出，不超过当年销售(营业)收入15%的部分，准予扣除；超过部分，准予在以后纳税年度结转扣除。

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
8

109

For the production links, the cost equations are linear with $C2 = 1$ for all links. Calculation using these cost parameters results in link costs with units of dollars per ton. As an example, the Central Pennsylvania production region has 12 million tons of low sulfur coal available for export. $C1$ is found by multiplying .086 times 12, resulting in 1.032. Since $C1$ must be an integer, it is rounded to 1. The link cost that results in the flow-dependent algorithm is $COST = 30 + 1 * 7 = 37$ \$/ton. Here the 7 is the link flow of 7 million tons per year on link 1-8.

The last link in the network, but the second set of links on the input menus, is the utility, and the link cost is based solely on the burn cost at the utility. This is a constant cost for each utility, but the data is very sketchy to support any particular set of numbers for these values. The fixed cost can be set by the user in the input menu, or the default of \$2.00 can be used for the link. $C1=0$ and $C2=1$ for all of these utility links.

The link costs for the US truck links are based on mileage from the mine to the tipple and on the size of the truck used (see menu 3). These cost equations are also linear with $C2 = 0$, and there is no fixed cost element so $C0 = 0$ also. The equation for the multiplier, $C1$, is $Cost/Ton = a + b * (distance)/(trucksize)$. The default values for the elements in this equation are $a=.50$ and $b=1.50$. The user can change these values on the menu for parameters in the input routine. The trucksize has been fixed at 20 tons and is not a variable which the user can change, although changes in can be reflected in "b".

| FOREIGN UTILITY INPUT FILE \

UTILITY	ORIG	DEST		RATE OF	MAX	BURN	SULF	LEAD	GENER	
NAME	NODE	NODE	COST	PRICE	USE	USE	COST	PREF	TIME	RATE
TEMPERE	74	84	2		2	4				
HANOVER	75	85	2		2	4				
APELDOOR	76	86	2		4	8				
FRANKFUR	77	87	2		4	8				
THAMES P	78	88	2		4	8				
MADRID	79	89	2		3	6				
LYONS	80	90	2		5	10				
ROME	81	91	2		3	6				
MARRAKES	82	92	2		1	2				
KYOTO	83	93	2		4	8				

MENU 2

/ TRUCK INPUT FILE \

[illegible]**MENU 3**

The US rail links, like the production and truck links, exhibit diseconomies of scale, however, the marginal cost curves are not linear. Several factors are important in setting the cost of the rail links (see menu 4), among them: whether the charges are based on single cars or on unit trains; whether a single railroad handles the entire trip or there is an interchange; whether the link is single or double track and what type of signal system it has; what the density of total traffic on the link is; and how long the haul on the link is. All of these factors are input by the user on the rail link menu. The coefficients are taken from a series of cost vs volume graphs which are attached to this Appendix. To simplify the calculations, the downward sloping portions of the curves at very low volumes are ignored.

For barge links, no data was available for the variable costs, but there was data on the total cost of representative shipments, and those are used here. Those links are entered on the transshipment menu, which is described below, as fixed cost links with $C1=0$ and $C2=1$.

The transshipment link costs are also considered to be fixed across any volume, and again $C1=0$ and $C2=1$. The fixed cost is based on the type of storage facility - railcar or ground; whether the facility can process different types of coal, and whether the facility services rail-to-ship or ship-to-ship (see menu 7). The full equation is $\text{Cost/Ton} = a + b*(\text{storage type}) + c*(\text{processing}) + d*(\text{loading type})$, and the coefficients are entered in the parameter menu.

The US port link will be used to model the government usage fees as they recapture their investment for dredging and for operation and

/ U.S. RAIL INPUT FILE

ROUTE	ORIG	DEST	TRANS	LINK	VAR-	SERV	INTR	LINK:				
NAME	NODE	NODE	COST	PRICE	DIST	ANCE	TIME	SIZE	TYPE	LINE	VOL	QUAL
RR 1	13	19			100				1	0	35	5
RR 2	13	20			396				1	0	25	2
RR 3	13	21			330				1	0	35	1
RR 4	13	22			330				1	0	40	2
RR 5	13	23			515				1	0	35	4
RR 6	14	19			308				1	0	30	4
RR 7	14	20			180				1	0	35	5
RR 8	14	21			350				1	0	40	5
RR 9	14	22			450				1	0	25	2
RR 10	14	23			678				1	0	25	1
RR 11	15	19			435				1	0	15	1
RR 12	15	20			410				1	0	10	1
RR 13	15	22			786				1	0	20	2
RR 14	15	23			412				1	0	25	2
RR 15	16	19			450				1	0	35	5

MENU 4

/ U.S. RAIL INPUT FILE

ROUTE	ORIG	DEST	TRANS	LINK	VAR-				SERV	INTR	LINK:		
NAME	NODE	NODE	COST	PRICE	DIST	IANCE	TIME	SIZE	TYPE	LINE	VOL	QUAL	
RR 16	16	23			412				1	0	25	2	
RR 17	16	24			497				1	0	8	1	
RR 18	16	25			913				1	0	25	2	
RR 19	16	26			999				1	0	5	2	
RR 20	17	25			944				1	0	20	2	
RR 21	17	27			150				1	0	15	4	
RR 22	18	24			480				1	0	7	1	
RR 23	18	25			994				1	0	15	2	
RR 24	18	26			660				1	0	14	2	
RR 25	13	94			520				1	0	40	4	
RR 26	13	95			750				1	0	15	2	
RR 27	14	94			550				1	0	25	4	
RR 28	15	96			500				1	0	30	5	
RR 29	16	96			900				1	0	35	4	
RR 30	16	95			380				1	0	25	2	

MENU 5

/ U.S. RAIL INPUT FILE

ROUTE	ORIG	DEST	TRANS	LINK	VAR-	SERV	INTR	LINK:				
NAME	NODE	NODE	COST	PRICE	DIST	ANCE	TIME	SIZE	TYPE	LINE	VOL	QUAL
RR 31	16	97		999					1	0	15	1
RR 32	18	98		720					1	0	5	1
RR 33	94	22		1					1	0	100	5
RR 34	95	23		1					1	0	100	5
RR 35	96	19		1					1	0	100	5
RR 36	97	26		1					1	0	100	5
RR 37	98	24		1					1	0	100	5

MENU 6

● 2019年12月10日，中国银保监会发布《关于规范商业银行互联网贷款业务的通知》，规定互联网贷款业务应当坚持小额、分散的原则，合理控制贷款集中度，不得发放大额、抵押类互联网贷款。单户用于消费的个人互联网贷款余额不得超过10万元，个人消费类贷款期限原则上不超过3年，最长不得超过5年。商业银行发放互联网贷款应当符合以下要求：（一）贷款集中度管理，发放互联网贷款余额不得超过本行一级资本净额的10%；（二）期限匹配，互联网贷款期限应当与借款人的还款能力相匹配，不得发放没有还款来源的无期限贷款，不得发放无还本付息的贷款；（三）利率上限，互联网贷款年化利率不得超过国家规定上限，不得通过附加费用等方式变相提高利率；（四）资金用途，互联网贷款资金应当通过受托支付的方式支付给交易对象，不得发放流动资金贷款用于股权投资、固定资产投资、房地产开发、生产经营活动、农副产品收购、消费者信用等；（五）风险管控，商业银行应当建立互联网贷款全流程风险管理体系，强化贷前、贷中、贷后全流程风险管理，不得通过互联网平台发放信用贷款；（六）信息披露，商业银行应当通过适当方式向借款人充分披露互联网贷款业务相关风险，不得通过互联网平台发放信用贷款；（七）消费者权益保护，商业银行应当建立互联网贷款消费者权益保护机制，不得通过互联网平台发放信用贷款；（八）数据安全，商业银行应当建立互联网贷款数据安全管理制度，不得通过互联网平台发放信用贷款；（九）其他要求，商业银行应当遵守其他相关法律法规规定。

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. There are two vertical lines on each side, creating a central writing area and margins. The paper appears to be from a notebook or a standard sheet of stationery.

117

maintenance costs at the port (see menu 8). The maintenance costs for the port are spread over the total volume of shipping in the port, but coal will bear 90% of the capital costs of the dredging. The payback of the construction costs will be annualized using a 12% APR over a 50-year term. This procedure is in keeping with the assumptions made by the US House of Representatives (11). A 50% recovery of the dredging investment is assumed in the model, and this means that some subsidy to the shippers is built into the model. The recovery per centage is set on the parameter menu which is explained below. This procedure will develop a total cost to be raised in port fees by the coal shippers on a yearly basis (C1). This amount will be divided (C2=-1) by the total tonnage of coal shipped through the dredged port to arrive at the per ton cost. For the Great Lakes ports there will be a fee of \$2.00 per ton to reflect the St. Lawrence Seaway tolls. This toll will be entered as a fixed cost (C0), and there will be no additional charges due to the low maintenance costs at those ports.

The vessel link will be constant over all flows, and it will be determined based on the length of the voyage and the maximum ship size which can be used, limited by the smaller port (see menu 9). The cost equation into which these values are inserted is

$\text{Cost/Ton} = 749 + .065 \text{ NMIL} - .008 \text{ DWT}$. This cost is given in cents per ton, and the equation is a regression analysis done by Lipfert.

The above linear regression equation works well for the size range of the ports currently found on the US east coast, but when these ports are dredged to handle 100,000 ton vessels, the vessel

/ U. S. PORT INPUT FILE \

PORT	ORIG	DEST	LOADING		MAX SHIP		CAPT	OPER	TOTAL		
NAME	NODE	NODE	COST	PRICE	VAR.	SIZE	DRAFT	COST	COST	VOL	
TOLEDO	29	38	2			26	26		2	22	
ASHTABUL	30	39	2			26	26		2	7	
PHIL	31	40				61	40		4	3	
BALT	32	41				61	40		7	17	
H ROADS	33	42				94	55		6	54	
SAVANNAH	34	43				54	35		8	5	
MOBILE	35	44				61	40		9	9	
N ORLEAN	36	45				61	40		16	12	
NO RIVER	37	46				61	40		4	2	

MENU 8

/ VESSEL INPUT FILE

VESSEL	ORIG	DEST	TIME VAR-		SHIPMNT	LINK					
NAME	NODE	NODE	COST	PRICE	TIME	LANCE	SIZE	DIST	DRAFT	FLAG	
TOL 1	38	47			6	20	26	4300	26	1	
TOL 2	38	48			6	20	26	3950	26	1	
TOL 3	38	49			6	20	26	3600	26	1	
TOL 4	38	50			6	20	26	3800	26	1	
TOL 5	38	51			6	20	26	4350	26	1	
TOL 6	38	52			6	20	26	4860	26	1	
TOL 7	38	53			6	20	26	3800	26	1	
TOL 8	38	54			20	30	26	12310	26	1	
TOL 9	38	55			6	20	26	4330	26	1	
ASH 1	39	47			6	20	26	4180	26	1	
ASH 2	39	48			6	20	26	3830	26	1	
ASH 3	39	49			6	20	26	3480	26	1	
ASH 4	39	50			6	20	26	3680	26	1	
ASH 5	39	51			6	20	26	4220	26	1	
ASH 6	39	52			6	20	26	4740	26	1	

MENU 9

/ VESSEL INPUT FILE

VESSEL	ORIG	DEST	TIME VAR-		SHIPMNT	LINK				
NAME	NODE	NODE	COST	PRICE	TIME	LANCE	SIZE	DIST	DRAFT	FLAG
ASH 7	39	53			6	20	26	3680	26	1
ASH 8	39	54			20	30	26	12190	26	1
ASH 9	39	55			6	20	26	4110	26	1
PHIL 1	40	47			6	20	34	4550	32	1
PHIL 2	40	48			6	20	61	4040	40	1
PHIL 3	40	49			6	20	61	3540	40	1
PHIL 4	40	50			6	20	29	3480	29	1
PHIL 5	40	51			6	29	61	4030	40	1
PHIL 6	40	52			6	20	61	4550	40	1
PHIL 7	40	53			6	20	28	3490	27	1
PHIL 8	40	54			18	30	61	15260	40	1
PHIL 9	40	55			6	20	31	4010	31	1
BALT 1	41	47			6	20	34	4690	32	1
BALT 2	41	48			6	20	61	4190	40	1
BALT 3	41	49			6	20	61	3680	40	1

MENU 10

/ VESSEL INPUT FILE

VESSEL	ORIG	DEST	TIME VAR-		SHIPMNT	LINK				
NAME	NODE	NODE	COST	PRICE	TIME	LANCE	SIZE	DIST	DRAFT	FLAG
BALT 4	41	50			6	20	29	3660	29	1
BALT 5	41	51			6	20	61	4180	40	1
BALT 6	41	52			6	20	61	4710	40	1
BALT 7	41	53			6	20	28	3620	27	1
BALT 8	41	54			18	30	61	15310	40	1
BALT 9	41	55			6	20	31	4170	31	1
HR 1	42	47			6	20	34	4570	32	1
HR 2	42	48			6	20	90	4010	55	1
HR 3	42	49			6	20	94	3560	55	1
HR 4	42	50			6	20	29	3550	29	1
HR 5	42	51			6	20	94	4050	55	1
HR 6	42	52			6	20	94	4600	55	1
HR 7	42	53			6	20	28	3510	27	1
HR 8	42	54			17	30	94	15910	55	1
HR 9	42	55			6	20	31	4080	31	1

MENU 11

/ VESSEL INPUT FILE |

VESSEL	ORIG	DEST	TIME VAR-		SHIPMNT	LINK					
NAME	NODE	NODE	COST	PRICE	TIME	LANCE	SIZE	DIST	DRAFT	FLAG	
SAV 1	43	47			6	20	34	4880	32	1	
SAV 2	43	48			6	20	54	4370	35	1	
SAV 3	43	49			6	20	54	3860	35	1	
SAV 4	43	50			6	20	29	3770	29	1	
SAV 5	43	51			6	20	54	4270	35	1	
SAV 6	43	52			6	20	54	4820	46	1	
SAV 7	43	53			6	20	28	3730	27	1	
SAV 8	43	54			20	30	54	15650	35	1	
SAV 9	43	55			6	20	31	4270	31	1	
MOB 1	44	47			6	20	34	5790	32	1	
MOB 2	44	48			6	20	61	5290	40	1	
MOB 3	44	49			6	20	61	4780	40	1	
MOB 4	44	50			6	20	29	4770	29	1	
MOB 5	44	51			6	20	61	5280	40	1	
MOB 6	44	52			6	20	61	5820	40	1	

MENU 12

/ VESSEL INPUT FILE |

VESSEL	ORIG	DEST	TIME VAR-		SHIPMNT	LINK					
NAME	NODE	NODE	COST	PRICE	TIME	LANCE	SIZE	DIST	DRAFT	FLAG	
MOB 7	44	53			6	20	28	4730	27	1	
MOB 8	44	54			18	30	61	15600	40	1	
MOB 9	44	55			6	20	31	5300	31	1	
NO L1	45	47			6	20	34	5840	32	1	
NO L2	45	48			6	20	61	5330	40	1	
NO L3	45	49			6	20	61	4830	40	1	
NO L4	45	50			6	20	29	4830	29	1	
NO L5	45	51			6	20	61	5330	40	1	
NO L6	45	52			6	20	61	5880	40	1	
NO L7	45	53			6	20	28	4790	27	1	
NO L8	45	54			18	30	61	15650	40	1	
NO L9	45	55			6	20	31	5350	31	1	
NO R1	46	47			6	20	34	5840	32	1	
NO R2	46	48			6	20	61	5330	40	1	
NO R3	46	49			6	20	61	4830	40	1	

MENU 13

/ VESSEL INPUT FILE

VESSEL	ORIG	DEST	TIME VAR-		SHIPMNT	LINK				
NAME	NODE	NODE	COST	PRICE	TIME	LANCE	SIZE	DIST	DRAFT	FLAG
NO R4	46	50			6	20	29	4830	29	1
NO R5	46	51			6	20	61	5330	40	1
NO R6	46	52			6	20	61	5880	40	1
NO R7	46	53			6	20	28	4790	27	1
NO R8	46	54			18	30	61	15650	40	1
NO R9	46	55			6	20	31	5350	31	1

MENU 14

link costs become too small. The linear regression fails to consider the diminishing returns to scale of the dredging beyond that capacity. In the sensitivity analysis, two runs were executed with vessel costs established using the equation

$$\text{COST} = (12.50 + \text{distance} * .0006) * (\text{vessel size} / 26) ** (-0.6).$$

The distance is measured in nautical miles and the vessel size in thousands of deadweight tons, and the cost results in dollars per ton. This equation is not linear, and it takes into account the diminishing returns of increasing the port capacity. Both the original vessel link costs and these new vessel link costs are fixed costs between two ports. Tables with both sets of vessel link costs, each before and after dredging, are included in Appendix C.

The foreign port and the foreign transshipment link costs are both fixed costs because of inadequate data. For the foreign port fees, a fee schedule similar to the US fee schedule was used based on the channel depth of the port (see menu 15). In this way ports handling larger ships carry a lower fee per ton than ports which can only handle smaller ships. The foreign transshipment fee of \$3.00 per ton for every port was taken from Lipfert (3) (see menu 16). This fixed fee acknowledges, as Lipfert stated, that there is a weakness in the network cost data but there is no better data available.

The foreign rail cost data is based on the European Rail Rates table in Lipert. These cost equations are linear ($C_2=1$) and they have no fixed component ($C_0=0$). The cost multiplier is given by $\text{Cost/Ton} = 4.85 + .08 * (\text{distance})$. The distance is given

FOREIGN PORT INPUT FILE \

PORT	ORIG	DEST	LOADING			MAX SHIP		CAPT	OPER	TOTAL
NAME	NODE	NODE	COST	PRICE	VAR.	SIZE	DRAFT	COST	COST	VOL
HELSINKI	47	56	2			34	32			4
HAMBURG	48	57	1			90	55			11
ROTTERDM	49	58	1			250	68			21
CADIZ	50	59	2			29	29			6
MARSEILL	51	60	1			160	60			7
TARRANTO	52	61	1			110	57			7
CASABLAN	53	62	2			28	27			4
NAGOYA	54	63	1			150	70			20
GENERIC	55	64	2			31	31			10

MENU 15

/μFOREIGN TSHIPMENT INPUT FILE \

=====											
TSHIPMENT	ORIG	DEST	LOADING		VARI-	STOR.	PROCES-	SHIP	MOS		
NAME	NODE	NODE	COST	PRICE	RATE	ANCE	METHOD	SING	SHIP	OPER	
HELSINKI	56	65	3		5	1	1	1	0	12	
HAMBURG	57	66	3		10	1	1	1	0	12	
ROTTERDM	58	67	3		20	1	1	1	1	12	
CADIZ	59	68	3		5	1	1	1	0	12	
MARSEILL	60	69	3		15	1	1	1	0	12	
TARRANTO	61	70	3		15	1	1	1	0	12	
CASABLAN	62	71	3		5	1	1	1	0	12	
NAGOYA	63	72	3		15	1	1	1	0	12	
GENERIC	64	73	3		30	1	1	1	0		
=====											

MENU 16

in miles and the cost is in dollars (see menu 17).

In order to enhance the usability of the input menus, the data has been simplified as much as possible by making the units on each link screen consistent for all inputs. All distances are entered in miles except the vessel voyage lengths which are in nautical miles. Volumes are entered in millions of tons per year for the production rates, the rail link volumes, transshipment rates, the total volume at the ports, and the rates of use at the utilities. The maximum ship size at the ports and the shipment size of the vessels are given in thousands of deadweight tons. On the parameter menu, all monetary coefficients are given in dollar values except c on the vessel menu which is too small to be given in dollars and is given in cents. The user should remember to enter any fixed cost he may desire to use in dollars per ton. These values will be assigned to C_0 , and $C_1=0$ and $C_2=1$ will be assigned for the other parameters.

In each link the cost coefficients are the values that are passed to the network algorithm to compute the link costs and the flow patterns. The prices are computed and stored in the input routine, and they are held there for possible use in future expansions of the model. In every case the costs are converted to cents when they are passed to the network algorithm so that they can be passed as integers. In the network algorithm they are manipulated in dollars and output in dollars. Both the input and the output, then, are in dollars which eliminates any data manipulation by the user.

For every link the user has the option of overriding the built-in

/ FOREIGN RAIL INPUT FILE |

ROUTE	ORIG	DEST	TRANS	LINK	VAR-	SERV	INTR	LINK:				
NAME	NODE	NODE	COST	PRICE	DIST	ANCE	TIME	SIZE	TYPE	LINE	VOL	QUAL
FR 1	65	74			120		2		1	0	4	1
FR 2	66	75			60		2		1	0	4	1
FR 3	67	76			100		2		1	0	8	1
FR 4B	67	77			325		4		1	0	8	1
FR 5B	67	78			250		3		1	0	6	1
FR 6	68	79			400		5		1	0	6	1
FR 7	69	80			220		5		1	0	10	1
FR 8	70	81			290		4		1	0	6	1
FR 9	71	82			160		2		1	0	2	1
FR10	72	83			100		2		1	0	6	1
FR11	73	77			120		2		1	0	2	1
FR12	73	78			30		1		1	0	2	1
FR13	73	81			60		1		1	0	2	1
FR14	73	83			20		1		1	0	2	1

MENU 17

cost equations by entering his own link costs on the link menus. This will enter a fixed cost into the network for the given link. An example of that is the barge link (27-28) which is in the network that was used for this study.

A second option for the user is to change the parameters used to calculate the cost coefficients. These parameters are entered on the parameter menu at the start of the input routine (see menu 18) This is one of the methods used to change the relevant data from the base case for the runs in this study. Care must be taken because these parameters affect every link entered on the input screen which they control, and changes can have large-scale impacts. These changes, however, give the user a powerful tool for experimenting with the network and evaluating different combinations of changes on the links.

/ MODEL PARAMETER FILE \

=====						
Input File	Cost Parameters:				Capacity	Cost Coefficient
Name:	A	B	C	D	Multiplier	C2
U.S. Mine	0	1.100	0			1
Truck	0.50	1.500				
U.S. Rail	5.00	1.100	4.00	.016		3.00
U.S. Transshipment	7.00	0	1.00	1.00	1.0	
U.S. Port	0	0.500				-1
Vessel	7.50	.0006	-8.00			
Foreign Port	1.00					
For. Transshipment	0	0	0	0		
Foreign Rail	4.85	.0800	4.00			
Utility						
Number of iterations:	2				Move size number:	1
Convergence criterion:	.020					
=====						

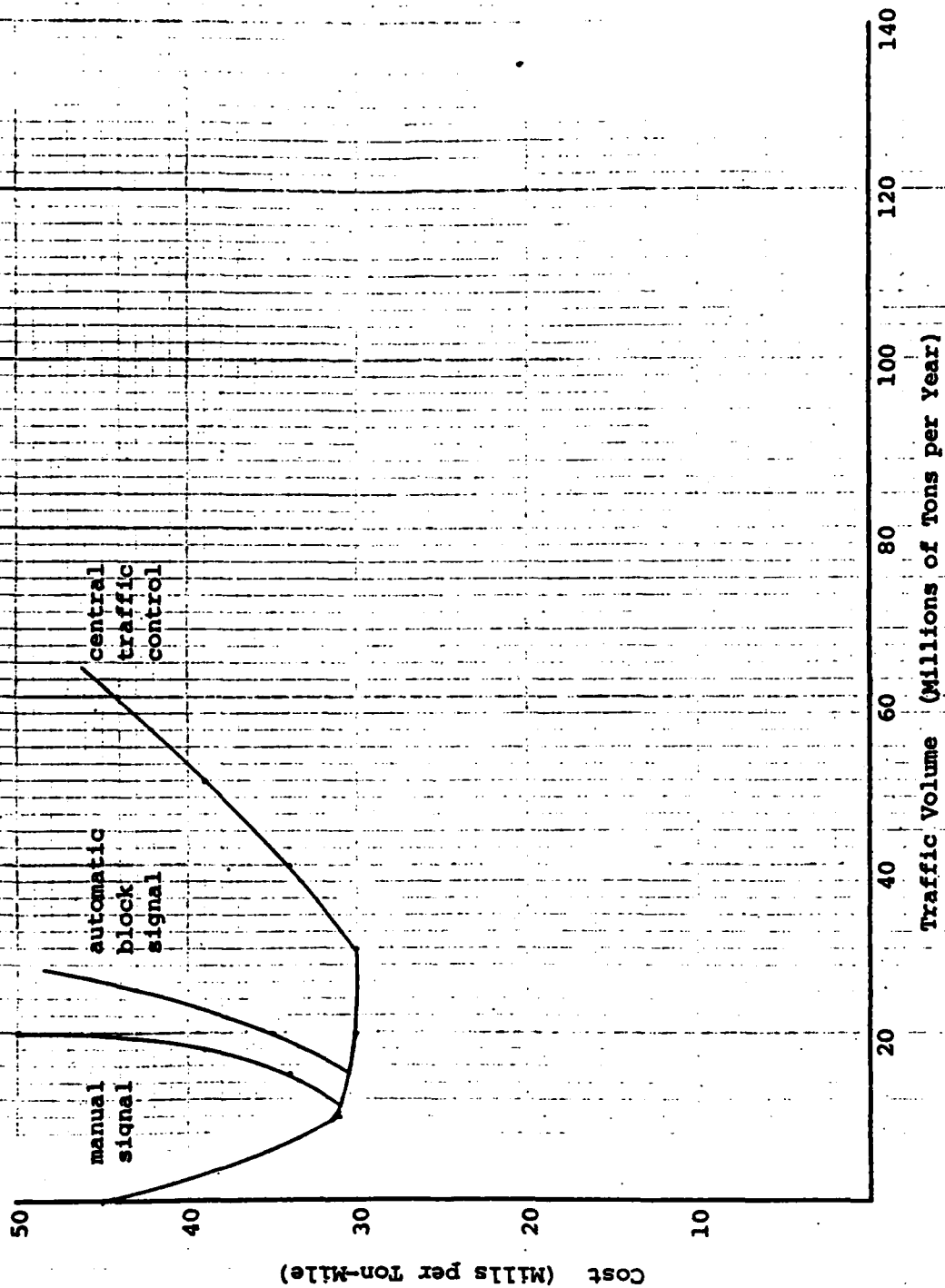
MENU 18

/ O-D FLOW MATRIX

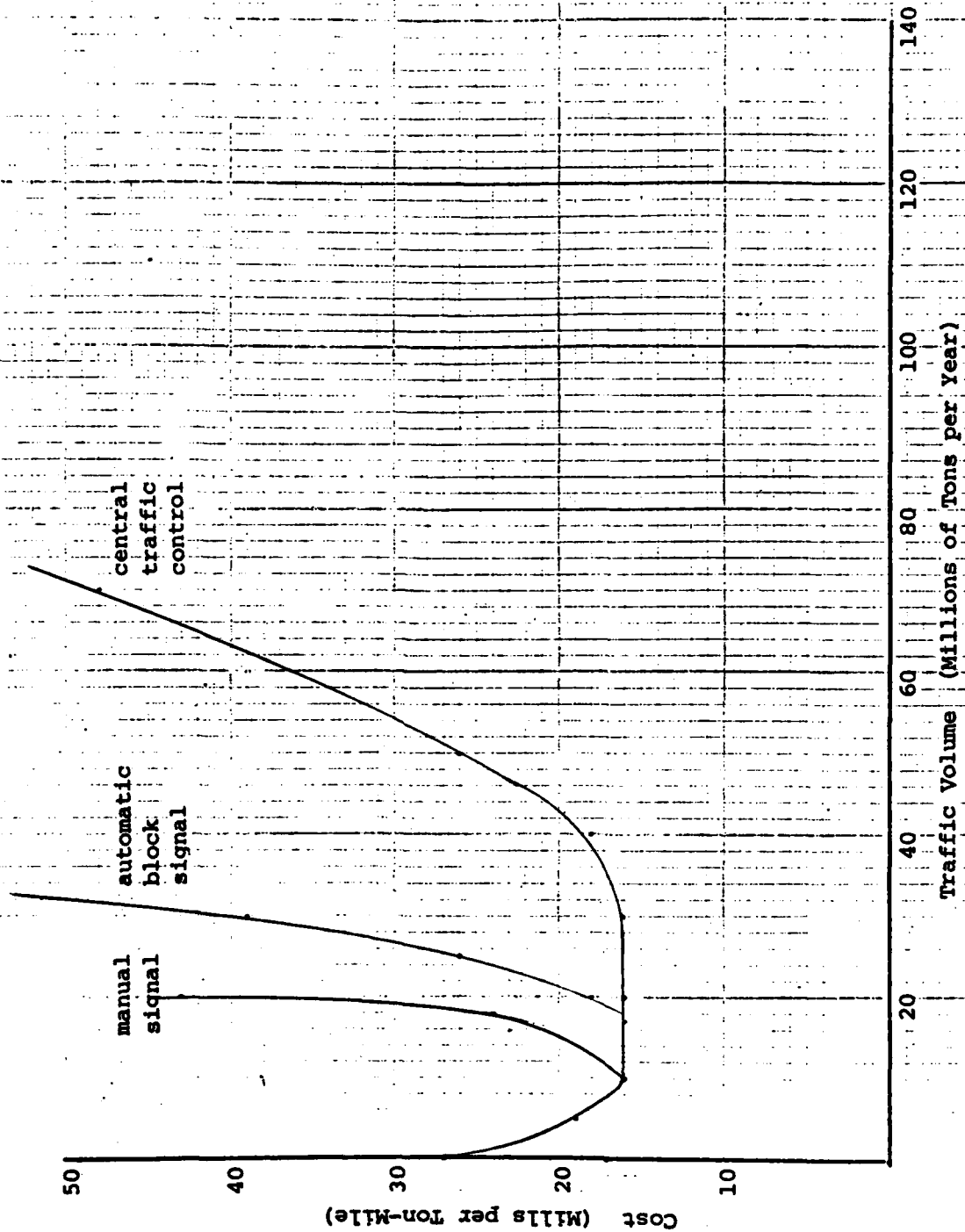
ORG	DESTINATION NUMBER									
#	84	85	86	87	88	89	90	91	92	93
1	1	0	0	3	3	0	0	0	0	0
2	0	1	1	1	3	0	0	0	1	0
3	0	1	0	2	2	0	0	1	0	0
4	0	0	0	1	2	1	1	0	0	1
5	0	1	0	0	0	1	0	0	0	1
6	0	0	0	0	0	0	2	0	0	1

MENU 19

Railroad Coal Cost Functions for Eastern Region Single Car Shipments (Single Track Lines)

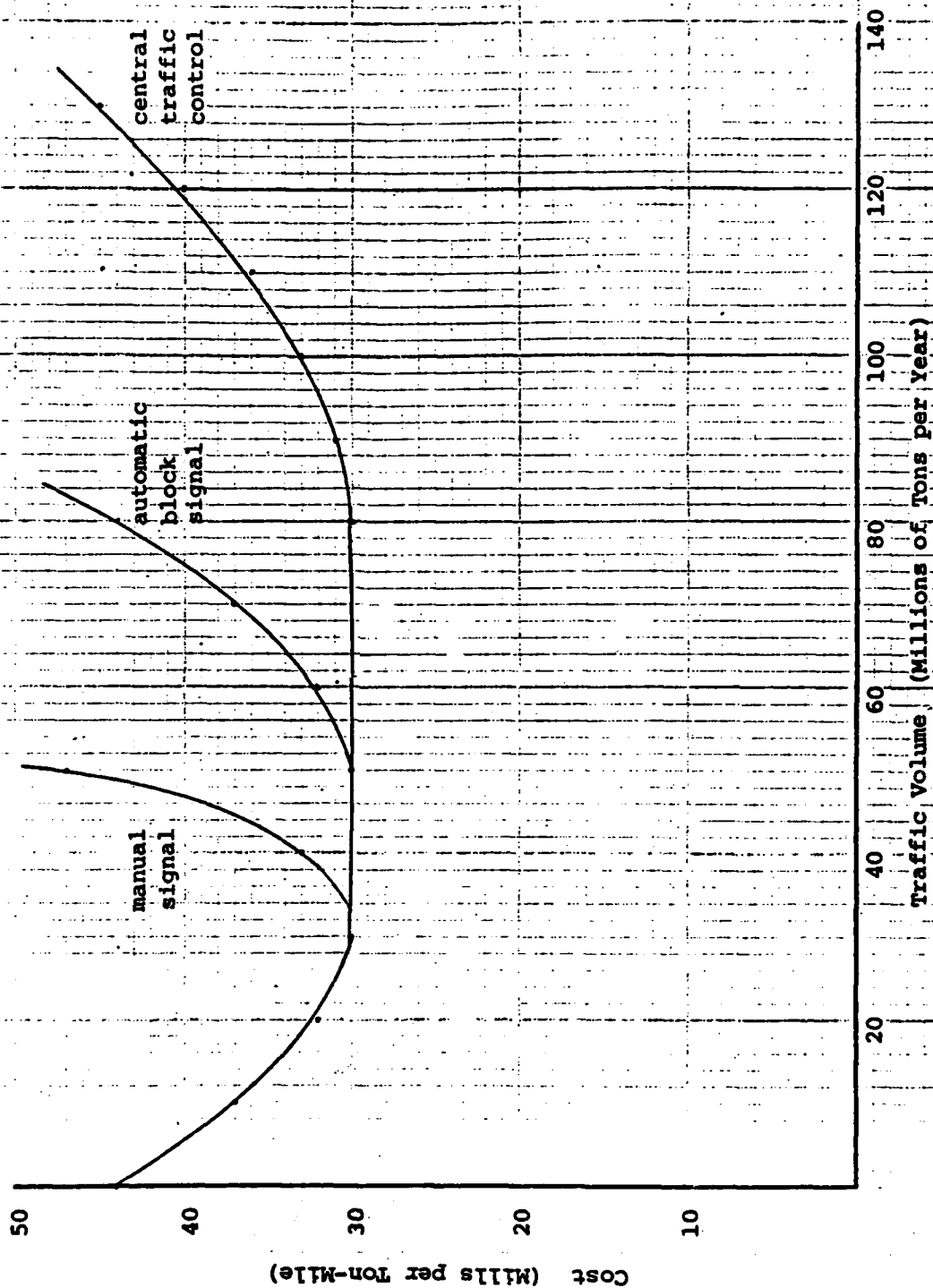


Railroad Coal Cost Functions for Eastern Region Unit Trains
(Single Track Lines)



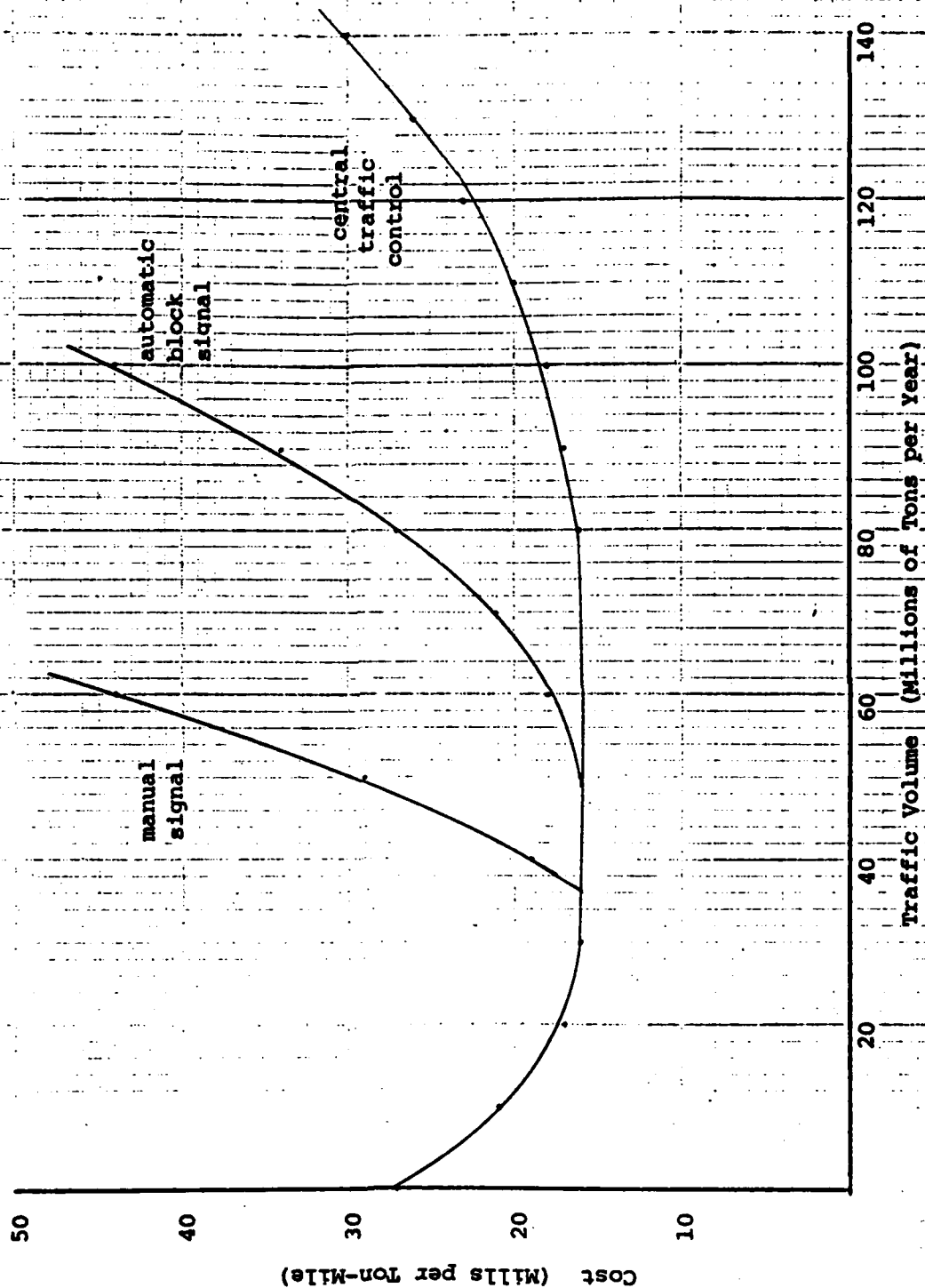
Railroad Cost Functions for Eastern Region Single Car Shipments

(Double Track Lines)



Railroad Cost Functions for Eastern Region Unit Trains

(Double Track Lines)

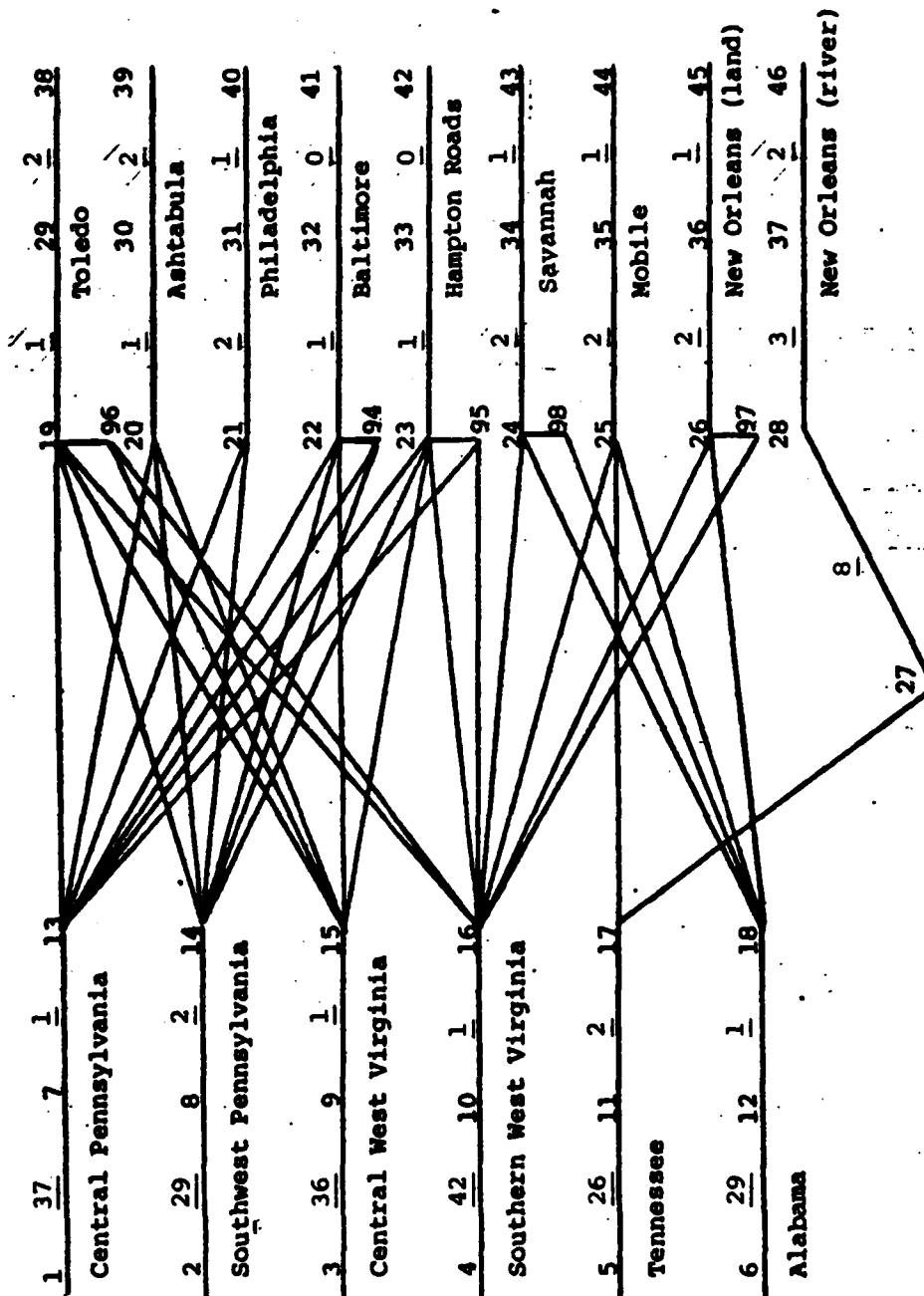


APPENDIX C LINK COST COEFFICIENTS - FLOW-DEPENDENT ALGORITHM

This Appendix contains a complete set of the link costs obtained in this study for the Base Case scenario (scenario 1) and for the scenario when Baltimore is dredged (scenario 2). The link costs for all links except the US rail links and the vessel links are displayed on network diagrams. The figure in the middle of the link which is underlined is the link cost, and the cost is in units of dollars per ton. For the rail link costs, the upper table of original rail link costs is for the Base Case, and the lower table shows the link costs from the scenario when Baltimore is dredged. The top table of original vessel link costs, similarly, displays the Base Case costs for all the vessel links. For scenario 2 when Baltimore is dredged, the costs for Baltimore (41) on the bottom must be substituted for the corresponding row of costs in the top table. All other vessel costs will remain the same as in the Base Case.

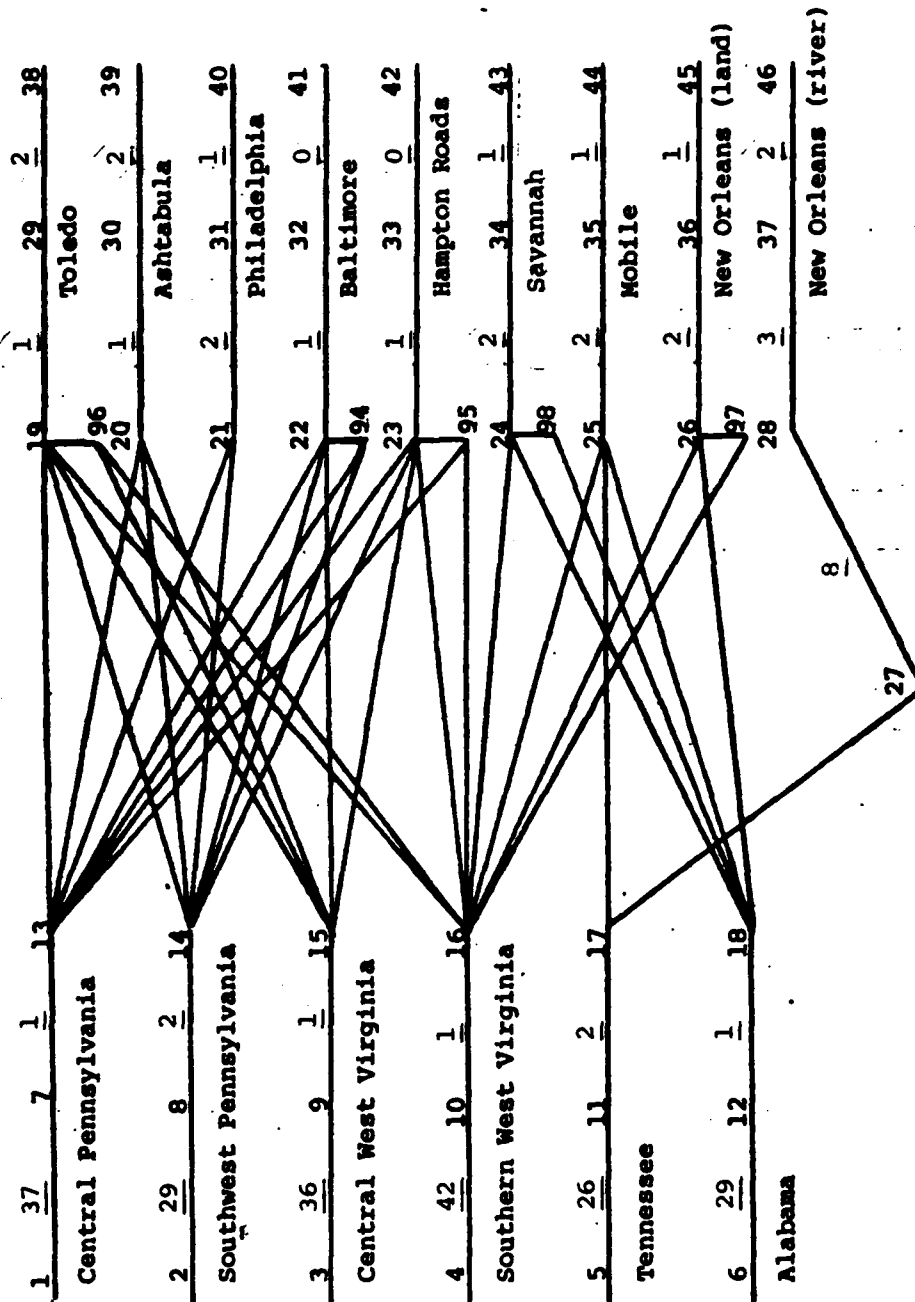
The last two pages of this Appendix show the rail prices and the new vessel link costs which were used in the second sensitivity analysis case (see section 4.6.2). The top table on each page refers to the base of the sensitivity analysis, and the bottom table displays the costs when Baltimore is dredged.

Export Coal Transportation Network in the United States



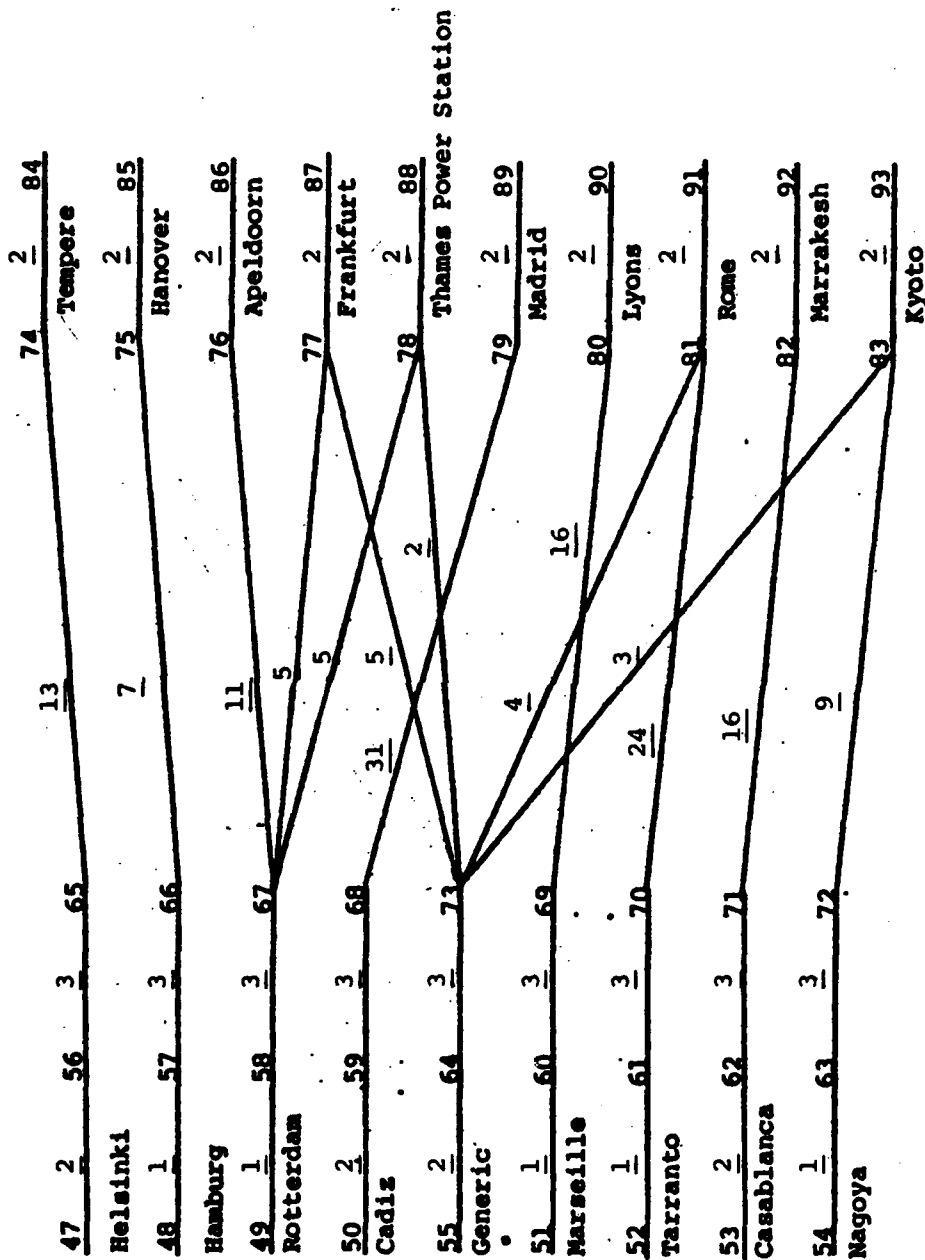
Base Case Link Costs
(\$/ Short Ton)

Export Coal Transportation Network in the United States

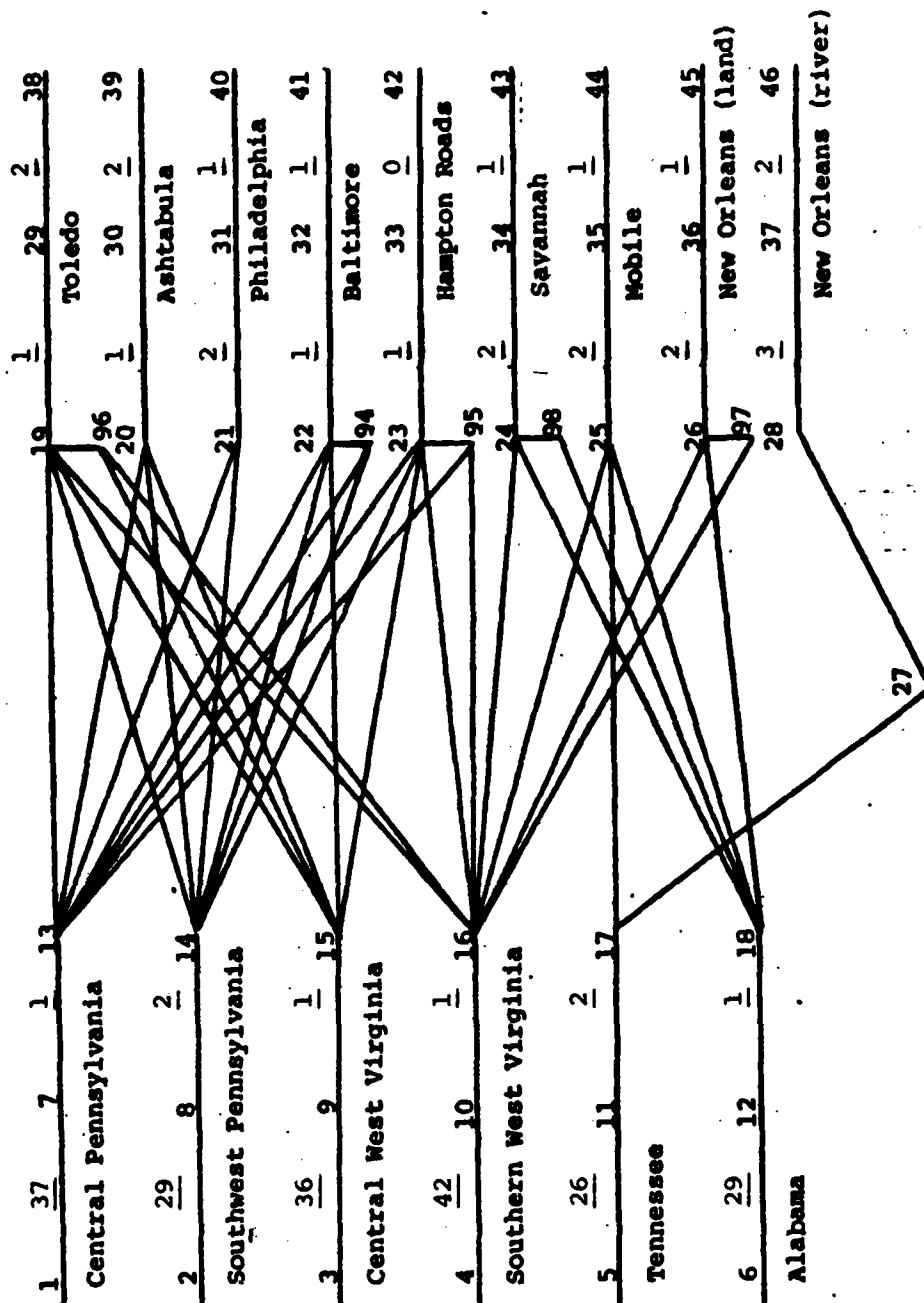


Base Case Link Costs

Export Coal Transportation Network
in Foreign Countries

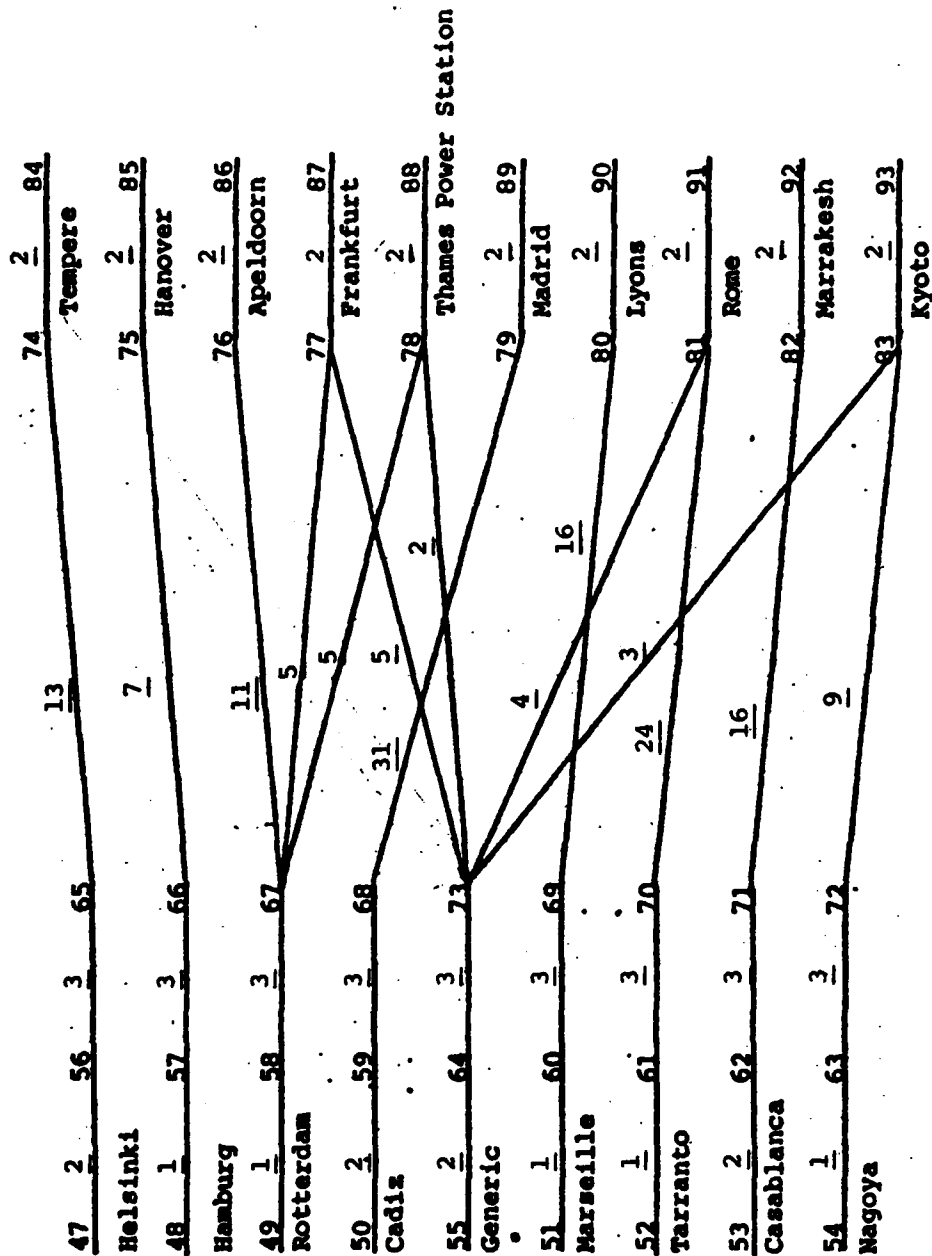


Export Coal Transportation Network in the United States



Link Costs Dredging Baltimore
(\$/Short Ton)

Export Coal Transportation Network in Foreign Countries



Link Costs Dredging Baltimore
(\$/Short Ton)

LINK COST MATRIX

ORIGINAL RAIL LINK COSTS

TO STATION	19	20	21	22	23	24	25	26	94	95	96	97	98
FROM STATION													
13	2	6	5	5	8				8	12			
14	5	3	6	7	11				9				
15	7	7		13	7						8		
16	7				7	8	15	24		6	14	29	
17							15						
18						8	16	11					12

Scenario 1

TO STATION	19	20	21	22	23	24	25	26	94	95	96	97	98
FROM STATION													
13	2	6	5	5	8				8	12			
14	5	3	6	7	11				9				
15	7	7		13	7						8		
16	7				7	8	15	24		6	14	29	
17							15						
18						8	16	11					12

Scenario 2

LINK COST MATRIX

ORIGINAL VESSEL LINK COSTS WITHOUT DREDGING

TO PORT	47	48	49	50	51	52	53	54	55
FROM PORT									
38	8	8	8	8	8	8	8	13	8
39	8	8	8	8	8	8	8	13	8
40	8	5	5	7	5	5	7	17	7
41	8	5	5	7	5	5	7	12	8
42	8	3	2	7	2	3	7	10	7
43	8	6	5	7	6	6	7	13	8
44	8	6	6	8	6	6	8	12	8
45	8	6	6	8	6	6	8	12	8
46	8	6	6	8	6	6	8	12	8

ORIGINAL VESSEL LINK COSTS WITH DREDGING

TO PORT	47	48	49	50	51	52	53	54	55
FROM PORT									
41	8	3	1	7	1	2	7	8	8
42	8	3	1	7	1	1	7	5	7
44	8	3	2	8	2	2	8	8	8

LINK COST MATRIX
MODIFIED RAIL LINK PRICES

TO STATION	19	20	21	22	23	24	25	26	94	95	96	97	98
FROM STATION													
13	10	5	14	15	19				21	15			
14	10	7	15	14	19				7				
15	8	7		13	12						8		
16	10				15	20	22	24		16	14	28	
17							18						
18						9	20	12					12

Scenario 19

TO STATION	19	20	21	22	23	24	25	26	94	95	96	97	98
FROM STATION													
13	10	5	14	15	19				21	15			
14	10	7	15	14	19				7				
15	8			13	12						8		
16	10				15	20	22	24		16	14	28	
17							18						
18						9	20	12					12

Scenario 20

LINK COST MATRIX

MODIFIED VESSEL LINK COSTS WITHOUT DREDGING

TO PORT	47	48	49	50	51	52	53	54	55
FROM PORT									
38	15	15	15	15	15	15	15	20	15
39	15	15	15	15	15	15	15	20	15
40	14	9	9	14	9	9	15	13	13
41	13	9	9	14	9	9	14	13	13
42	14	9	9	14	9	9	14	13	13
43	13	10	10	14	10	10	14	14	14
44	14	9	9	14	9	10	15	13	14
45	15	9	9	14	9	10	15	13	13
46	15	9	9	14	9	10	15	13	13

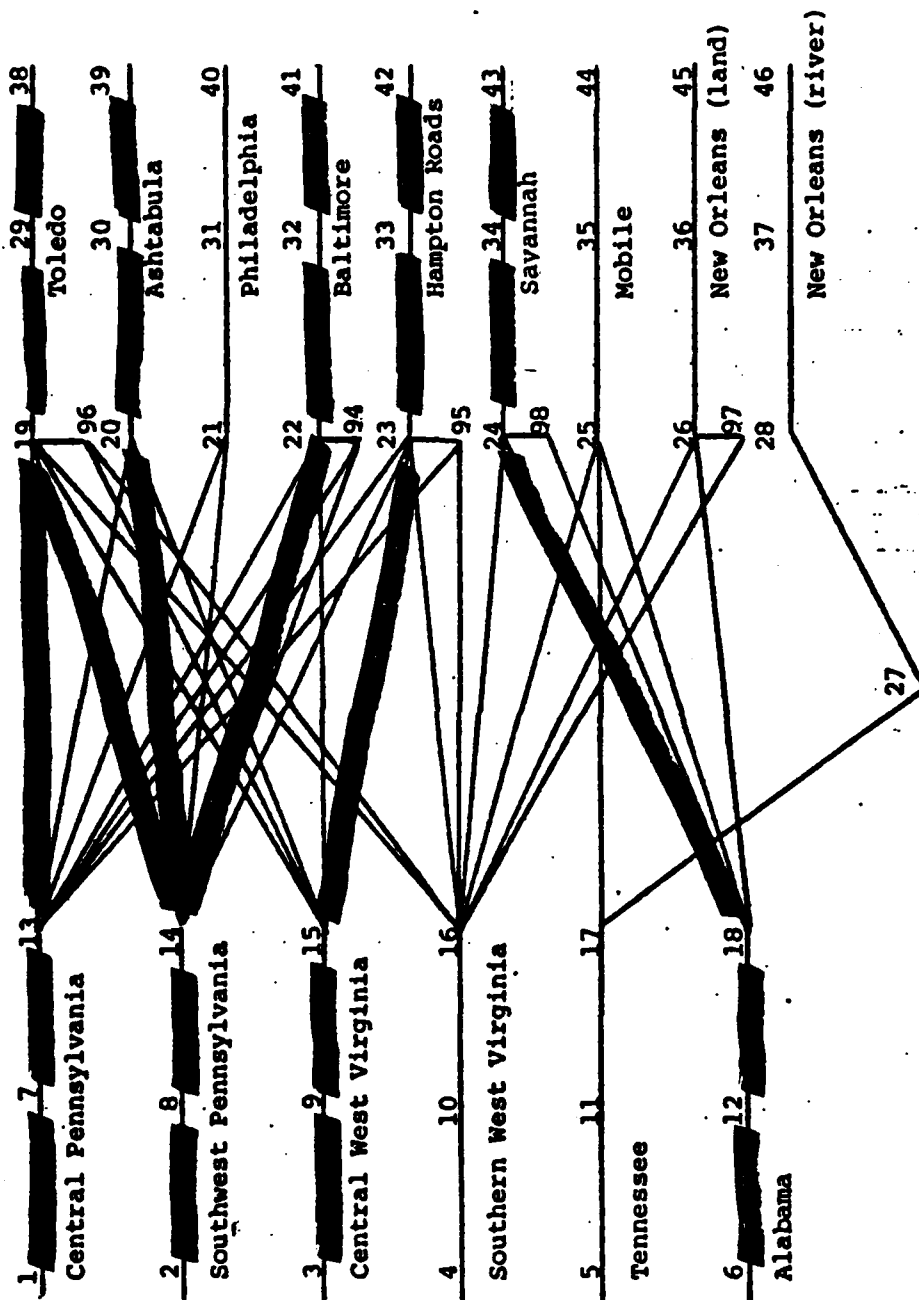
MODIFIED VESSEL LINK COSTS WITH DREDGING

TO PORT	47	48	49	50	51	52	53	54	55
FROM PORT									
41	13	7	6	14	6	6	14	9	13

APPENDIX D FLOW DIAGRAMS and O-D COST MATRICES

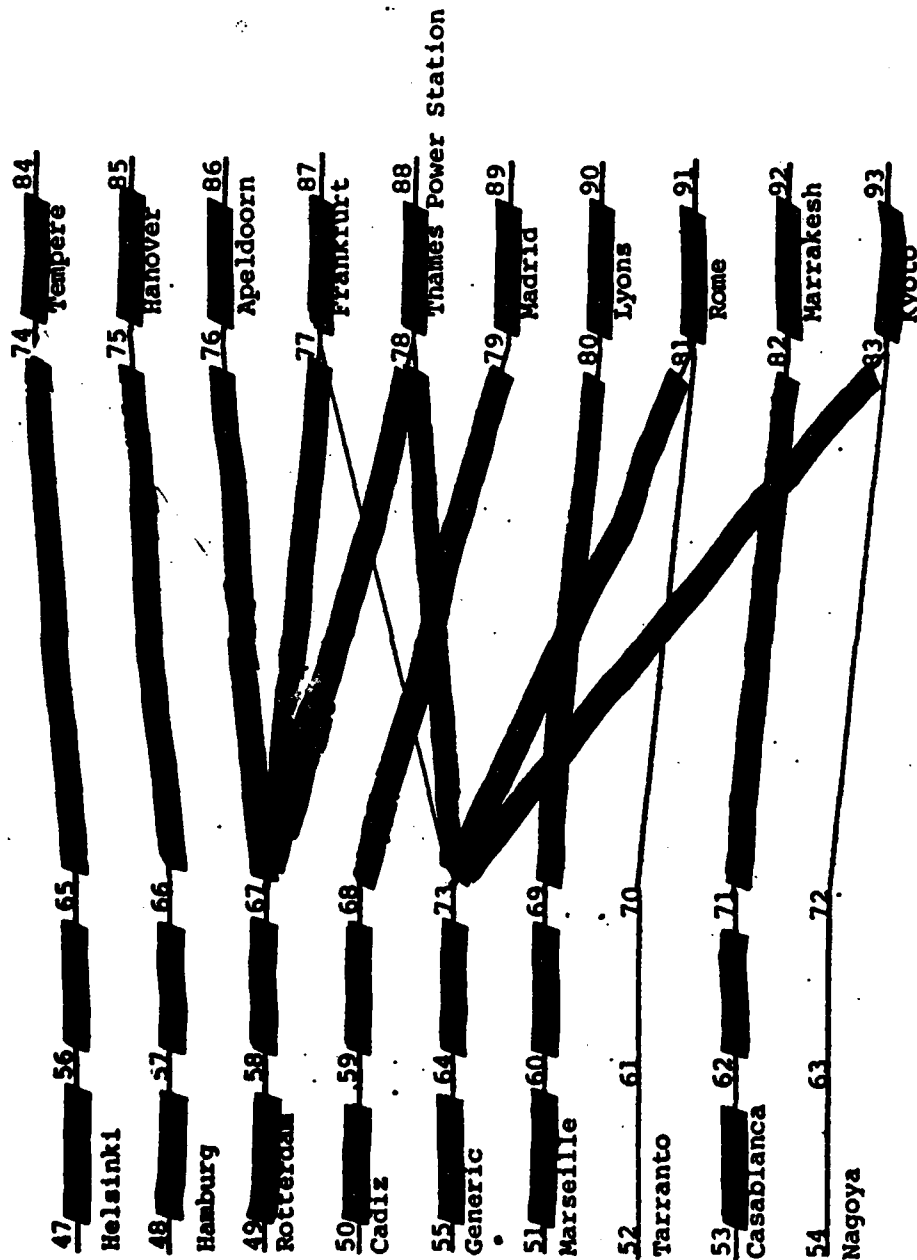
This Appendix contains the 20 sets of network diagrams showing the flow of export steam coal obtained by the model and the 20 cost matrices exhibiting the total path cost for each O-D pair. The cost matrix is output from the flow-dependent algorithm for every scenario, and the network diagrams show the flow patterns of the transshipment algorithm. The output for each scenario is presented in a set which consists of the transportation network in the US (labeled A), the transportation network network in the foreign countries (labeled B), and the O-D cost matrix (labeled C.) The first 16 sets are numbered to correspond with the 16 scenarios of the initial study of this thesis. Sets 17-20 depict the flows and costs of the sensitivity analysis.

Export Coal Transportation Network in the United States



4.4.1A Link Flows Base Case

Export Coal Transportation Network in Foreign Countries



4.4.1B Link Flows Base Case

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	69	62	66	60	58	87	71	60	72	59
2	63	56	60	54	52	81	65	54	66	53
3	73	61	64	58	58	90	69	63	75	62
4	79	67	70	64	64	96	75	69	81	68
5	73	64	68	62	62	91	73	64	76	63
6	68	59	62	56	56	85	68	59	70	58

4.4.1C Base Case Path Costs

1 ~~7~~ 13
Central Pennsylvania

2 ~~8~~ 14
Southwest Pennsylvania

3 ~~9~~ 15
Central West Virginia

4 10 16
Southern West Virginia

5 11 17
Tennessee

6 ~~12~~ 18
Alabama

7 19
Toledo

8 20
Ashtabula

9 21
Philadelphia

10 22
Baltimore

11 23
Hampton Roads

12 24
Savannah

13 25
Mobile

14 26
New Orleans (land)

15 27
New Orleans (river)

4.4.2A Link Flows Dredging Baltimore

47	██████████	56	██████████	65	██████████	74	██████████	84	██████████	Teapare
Helsinki										
48	██████████	57	██████████	66	██████████	75	██████████	85	██████████	Hanover
Hamburg										
49	██████████	58	██████████	67	██████████	76	██████████	86	██████████	Apeldoorn
Rotterdam										
50	██████████	59	██████████	68	██████████	77	██████████	87	██████████	Frankfurt
Cadiz										
55	██████████	64	██████████	73	██████████	78	██████████	88	██████████	Thames Power Station
Generic										
51	██████████	60	██████████	69	██████████	79	██████████	89	██████████	Madrid
Marseille										
52	██████████	61	██████████	70	██████████	80	██████████	90	██████████	Lyons
Tarranto										
53	██████████	62	██████████	71	██████████	81	██████████	91	██████████	Rome
Casablanca										
54	██████████	63	██████████	72	██████████	82	██████████	92	██████████	Marrakesh
Nagoya										
						83	██████████	93	██████████	Kyoto

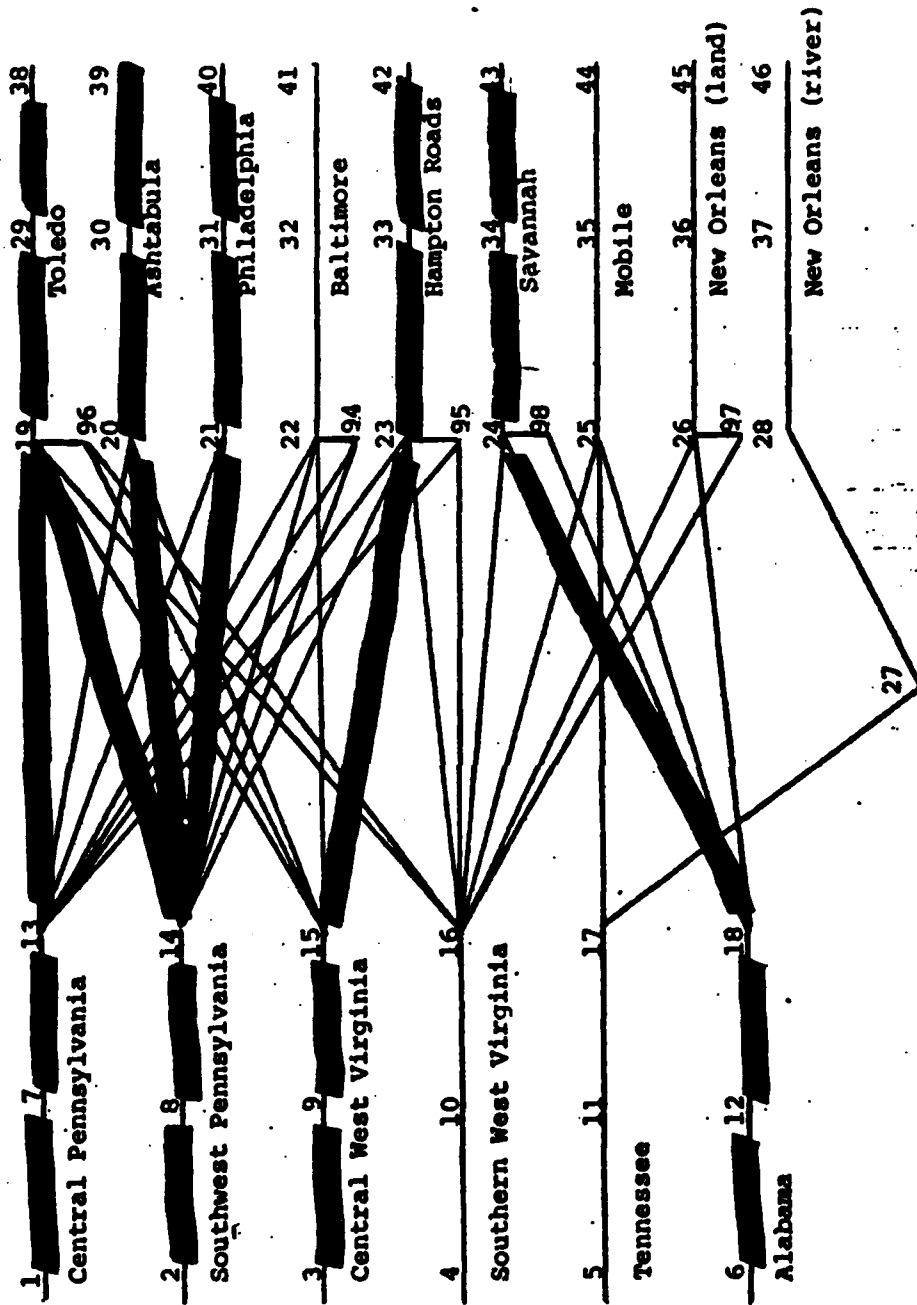
4.4.2B Link Flows Dredging Baltimore

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	71	60	62	56	56	89	67	62	74	61
2	65	55	57	51	51	83	62	56	68	55
3	73	61	64	58	58	90	69	63	75	62
4	79	67	70	64	64	96	75	69	81	68
5	73	64	68	62	62	91	73	64	76	63
6	68	59	62	56	56	85	68	59	70	58

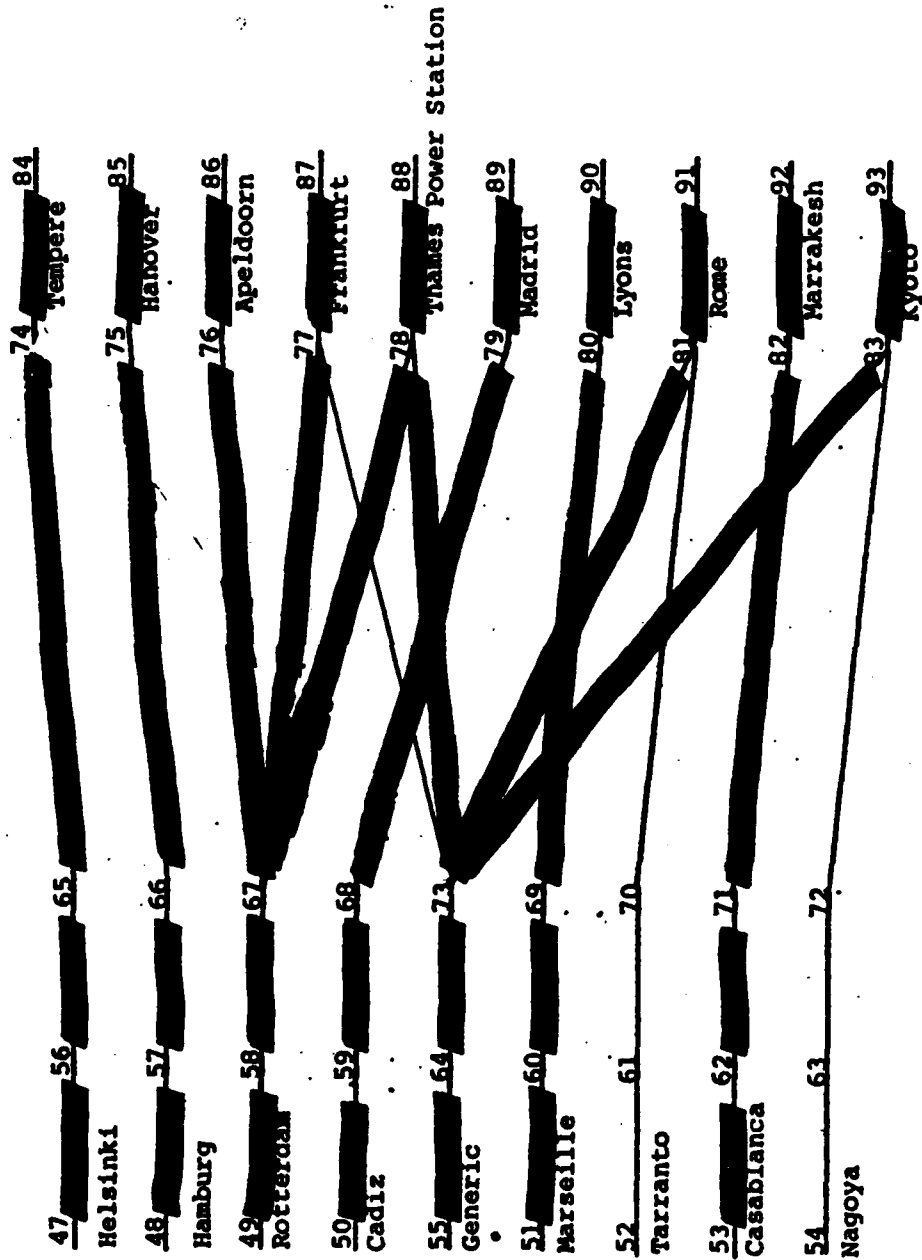
4.4.2C Path Costs Dredging Baltimore

Export Coal Transportation Network in the United States



4.4.3A Link Flows Dredging Hampton Roads

Export Coal Transportation Network
in Foreign Countries



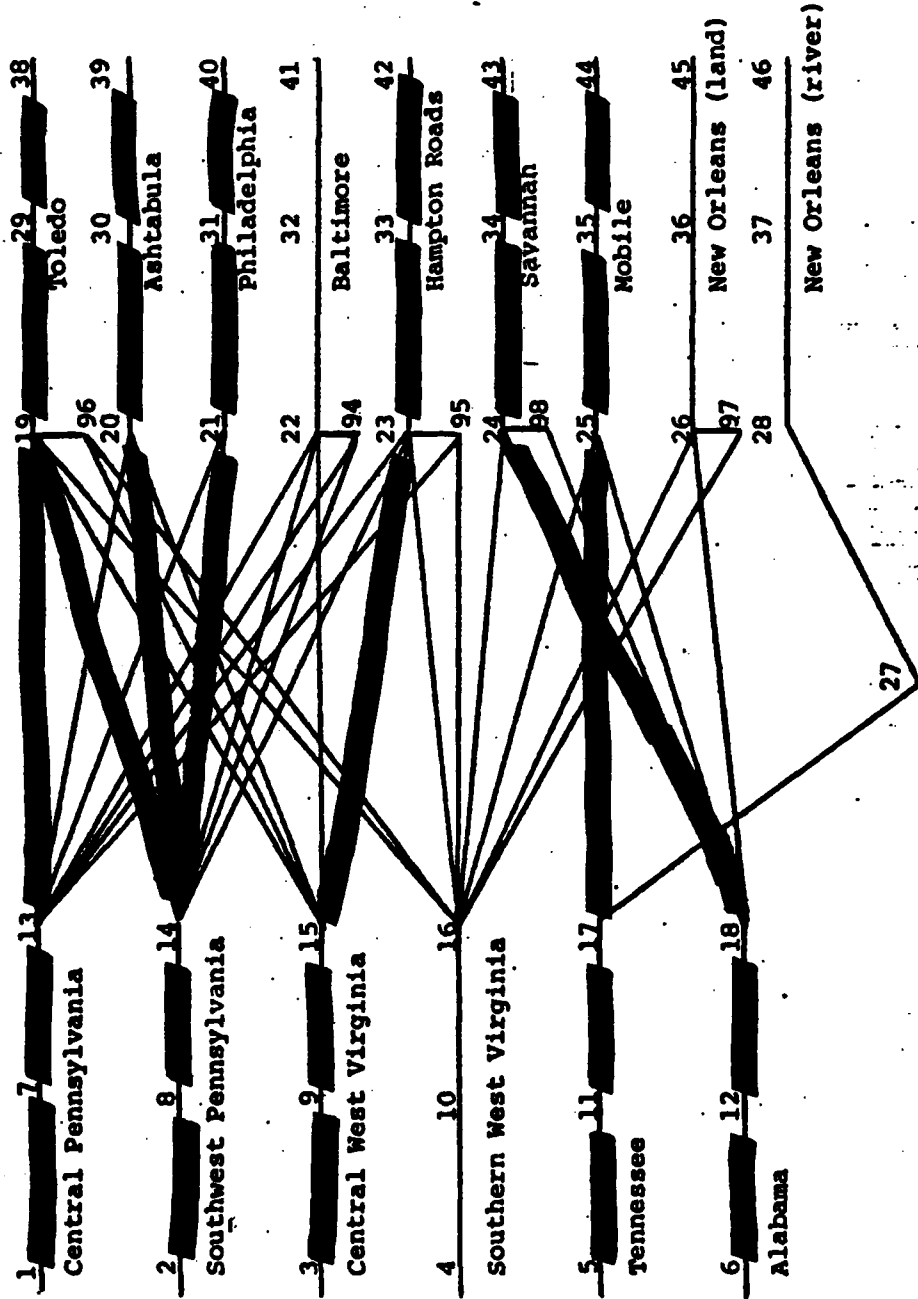
4.4.3B Link Flows Dredging Hampton Roads

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	69	62	65	59	58	87	70	60	72	59
2	63	56	60	54	52	81	65	54	66	53
3	73	61	63	57	57	90	68	63	75	62
4	79	67	69	63	63	96	74	69	81	68
5	73	64	68	62	62	91	73	64	76	63
6	68	59	62	56	56	85	68	59	70	58

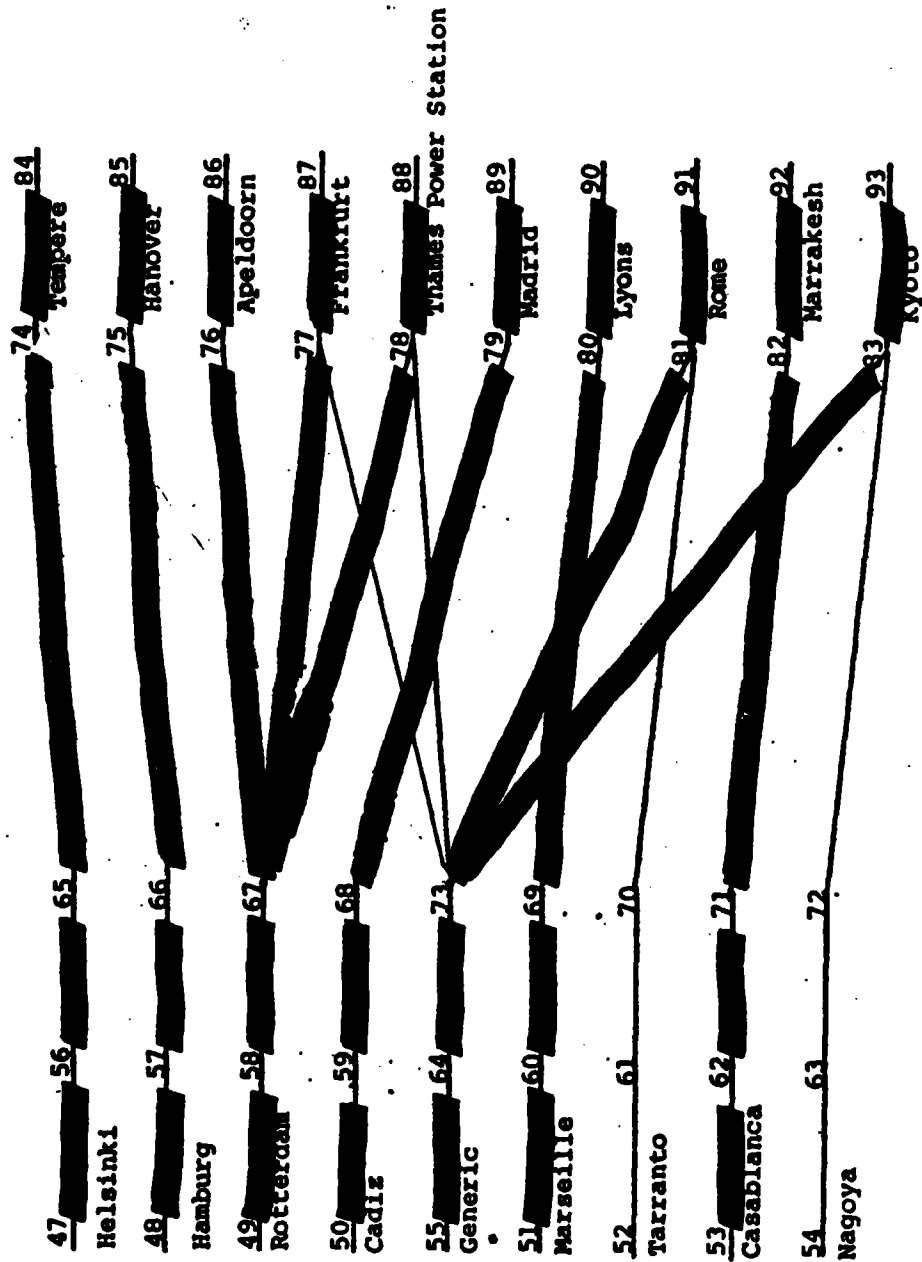
4.4.3C Path Costs Dredging Hampton Roads

Export Coal Transportation Network
in the United States



4.4.4A Link Flows Dredging Mobile

Export Coal Transportation Network
in Foreign Countries



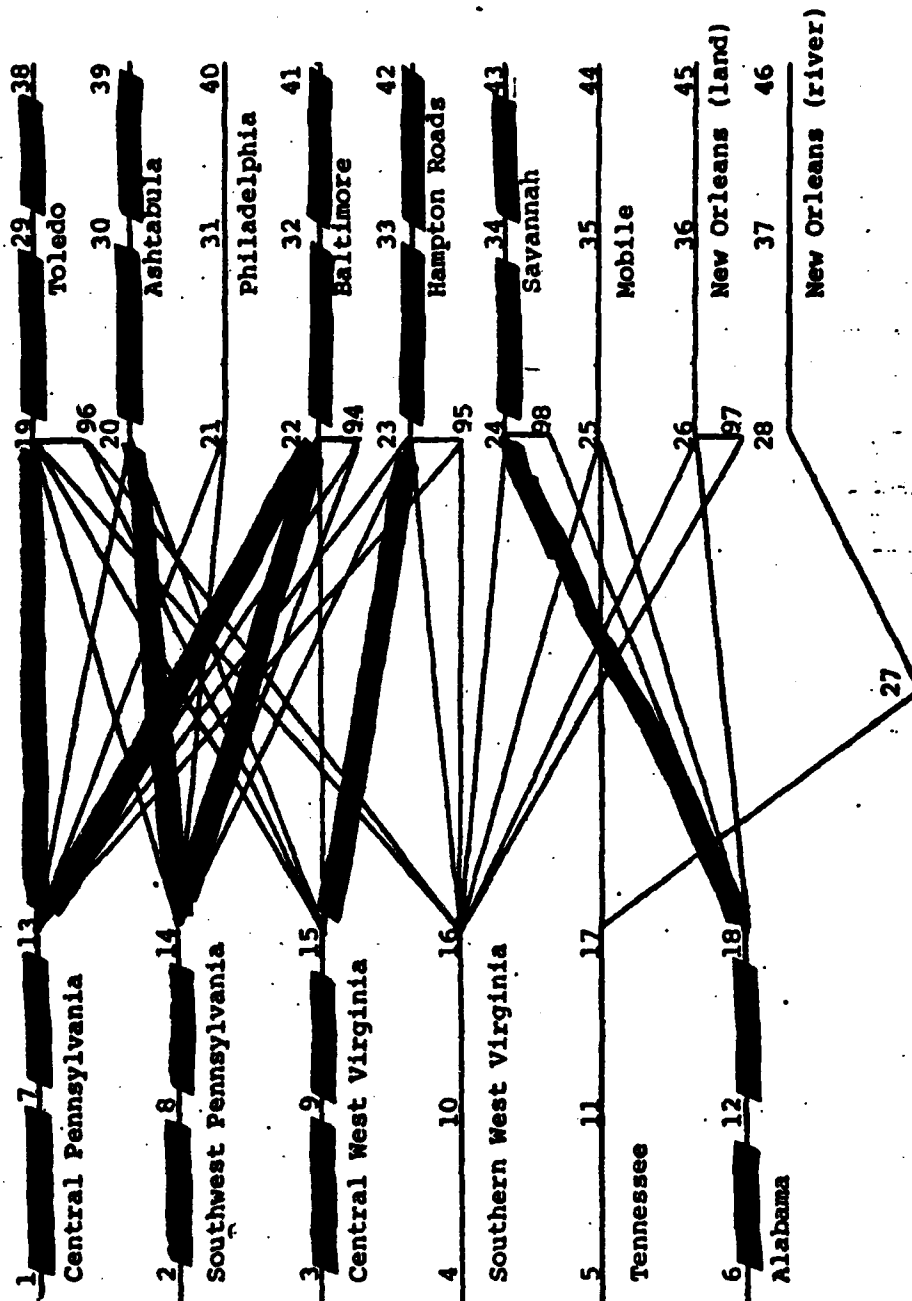
4.4.4B Link Flows Dredging Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	69	62	66	60	58	87	71	60	72	59
2	63	56	60	54	52	81	65	54	66	53
3	73	61	64	58	58	90	69	63	75	62
4	79	67	70	64	64	96	75	69	81	68
5	73	61	64	58	58	91	69	64	76	63
6	68	59	62	56	56	85	68	59	70	58

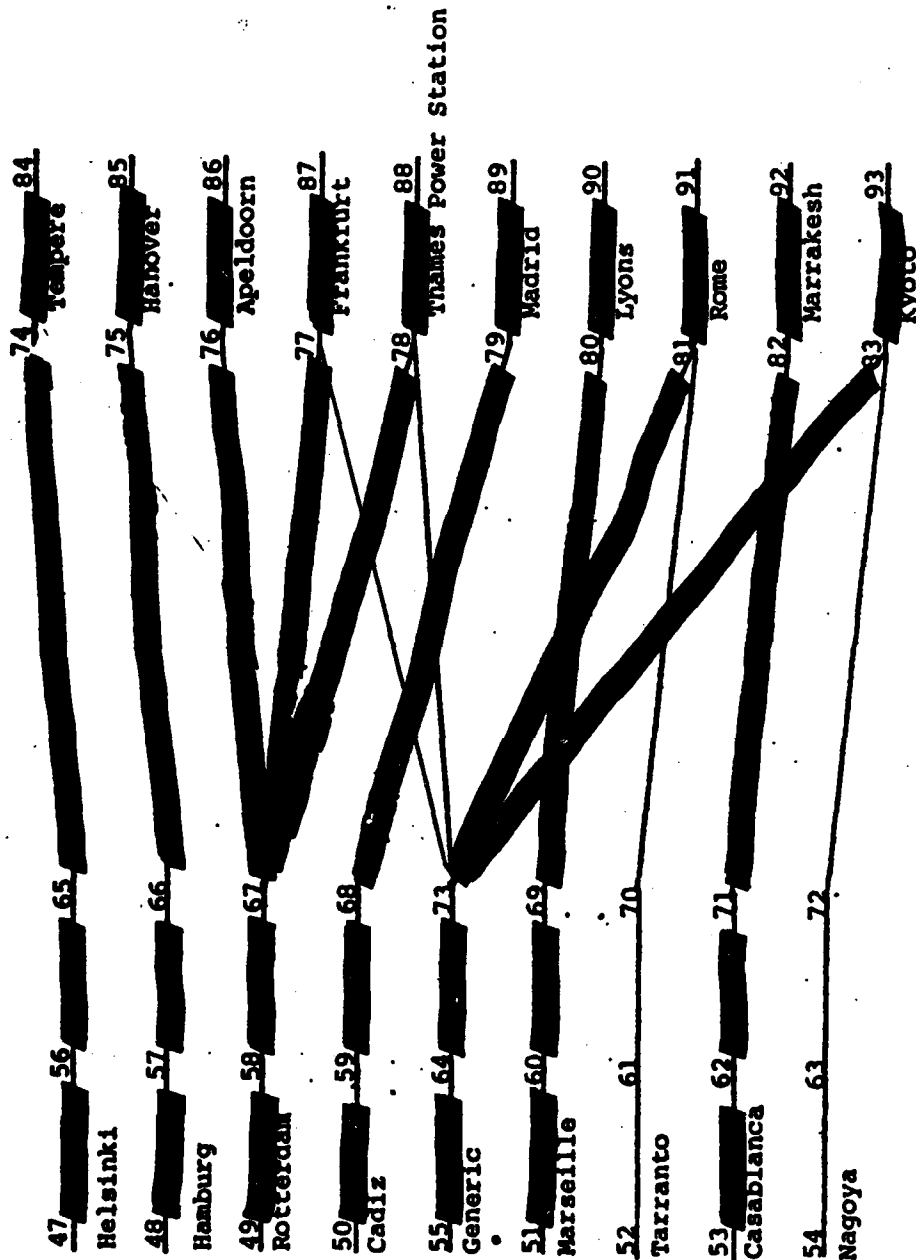
4.4.4C Path Costs Dredging Mobile

Export Coal Transportation Network
in the United States



4.4.5A Link Flows Dredging Baltimore and Hampton Roads

Export Coal Transportation Network in Foreign Countries



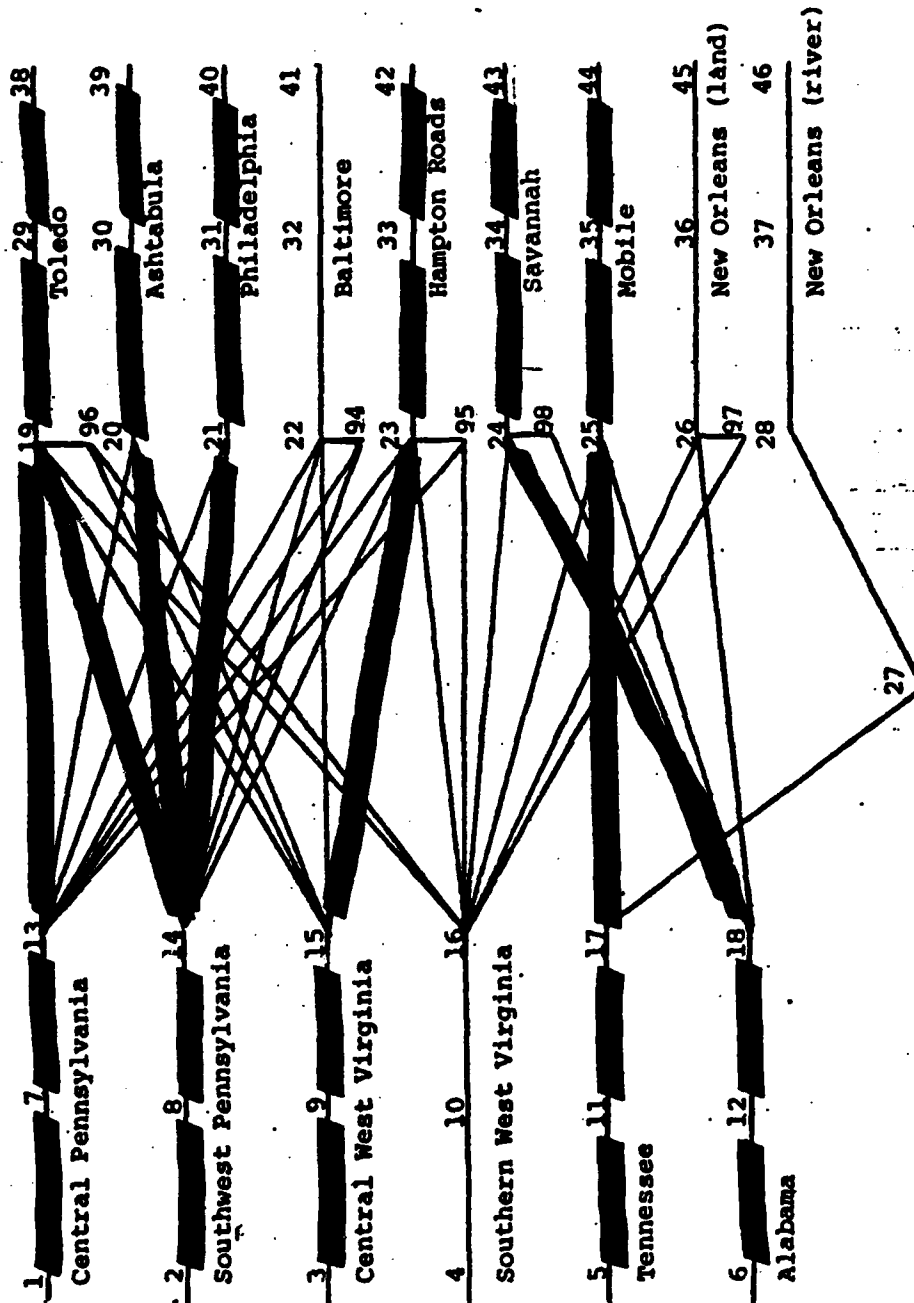
4.4.5B Link Flows Dredging Baltimore and Hampton Roads

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	71	60	62	56	56	89	67	62	74	61
2	65	55	57	51	51	83	62	56	68	55
3	73	61	63	57	57	90	68	63	75	62
4	79	67	69	63	63	96	74	69	81	68
5	73	64	68	62	62	91	73	64	76	63
6	68	59	62	56	56	85	68	59	70	58

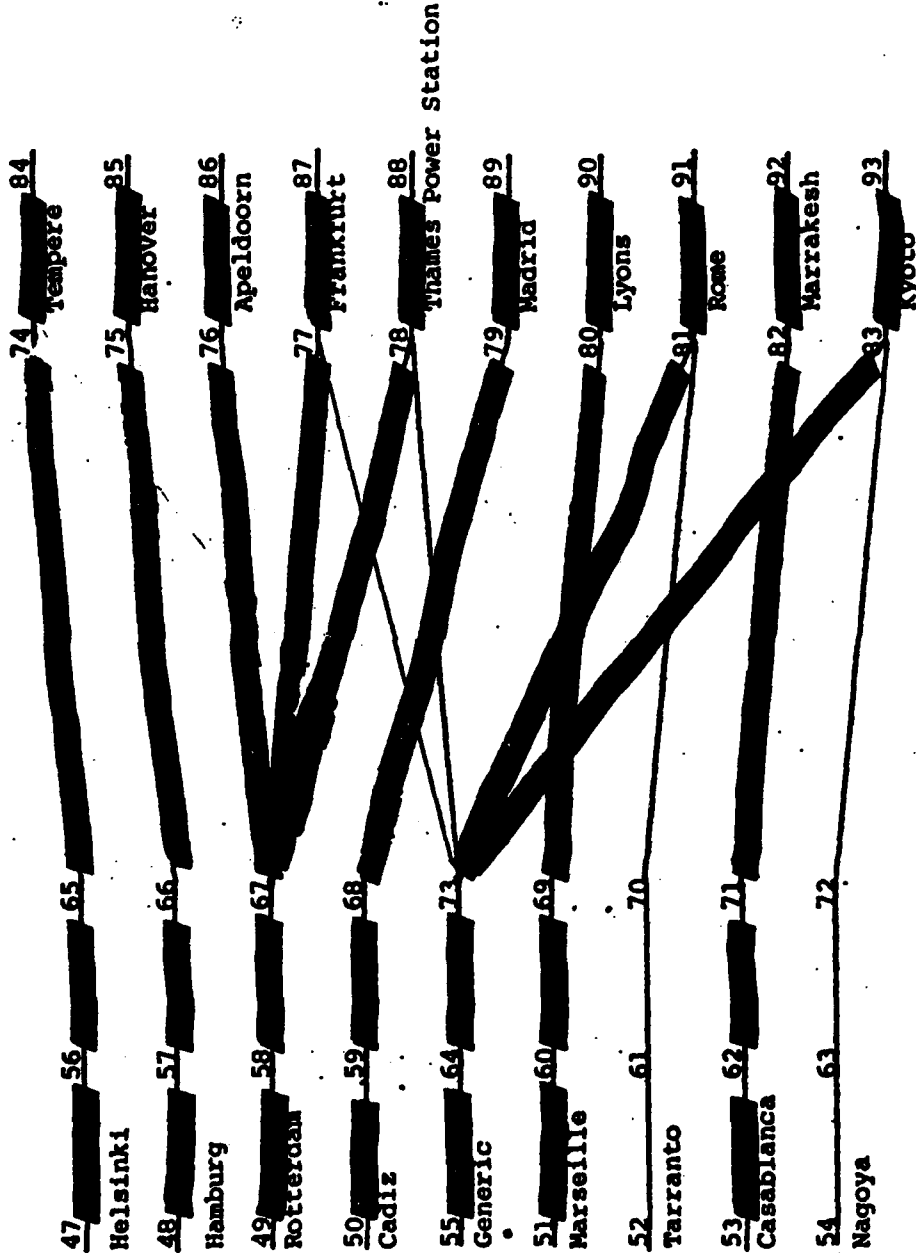
4.4.5C Path Costs Dredging Baltimore and Hampton Roads

Export Coal Transportation Network
in the United States



4.4.6A Link Flows Dredging Hampton Roads and Mobile

Export Coal Transportation Network
in Foreign Countries



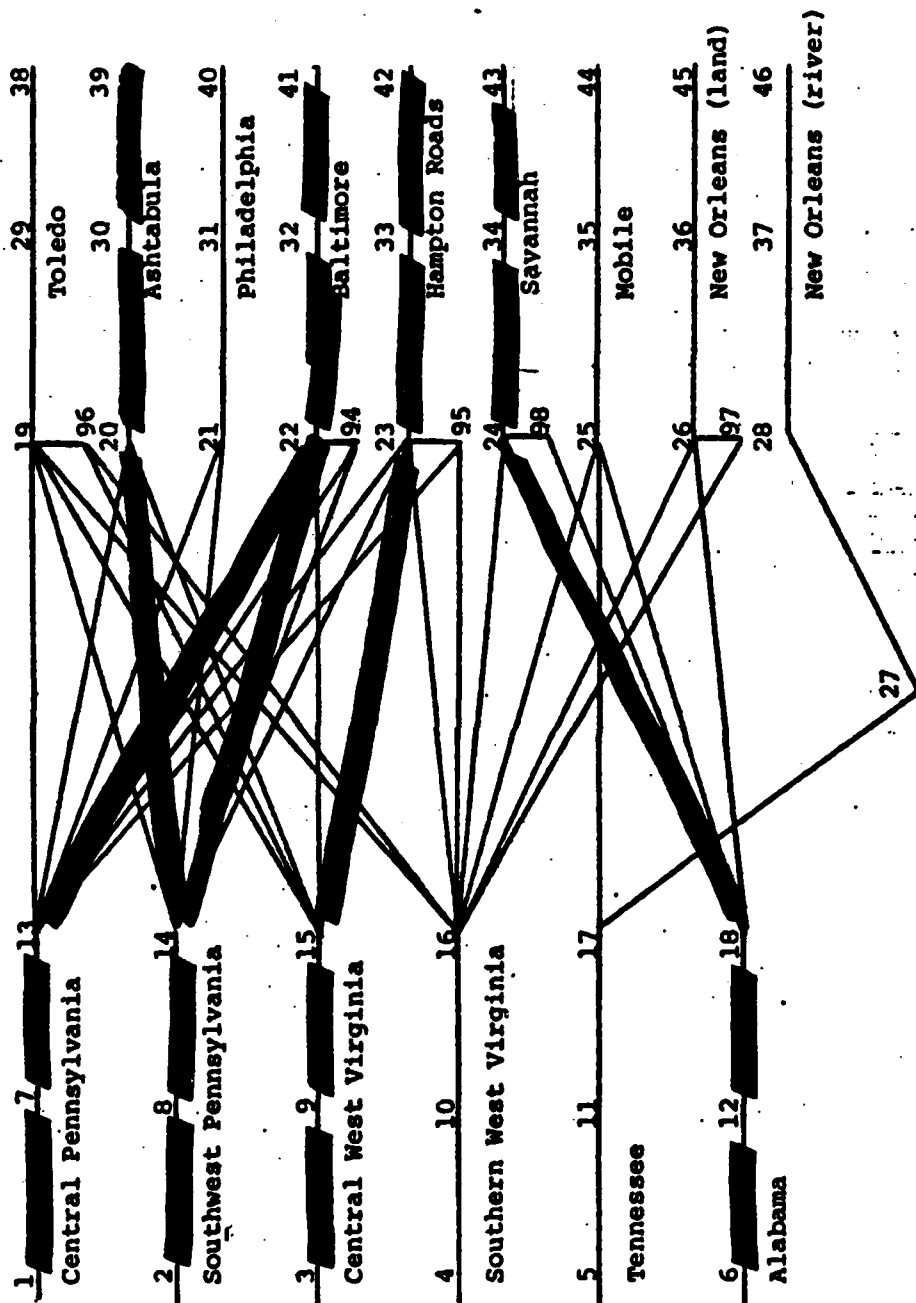
4.4.6B Link Flows Dredging Hampton Roads and Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	69	62	65	59	58	87	70	60	72	59
2	63	56	60	54	52	81	65	54	66	53
3	73	61	63	57	57	90	68	63	75	62
4	79	67	69	63	63	96	74	69	81	68
5	73	61	64	58	58	91	69	64	76	63
6	68	59	62	56	56	85	68	59	70	58

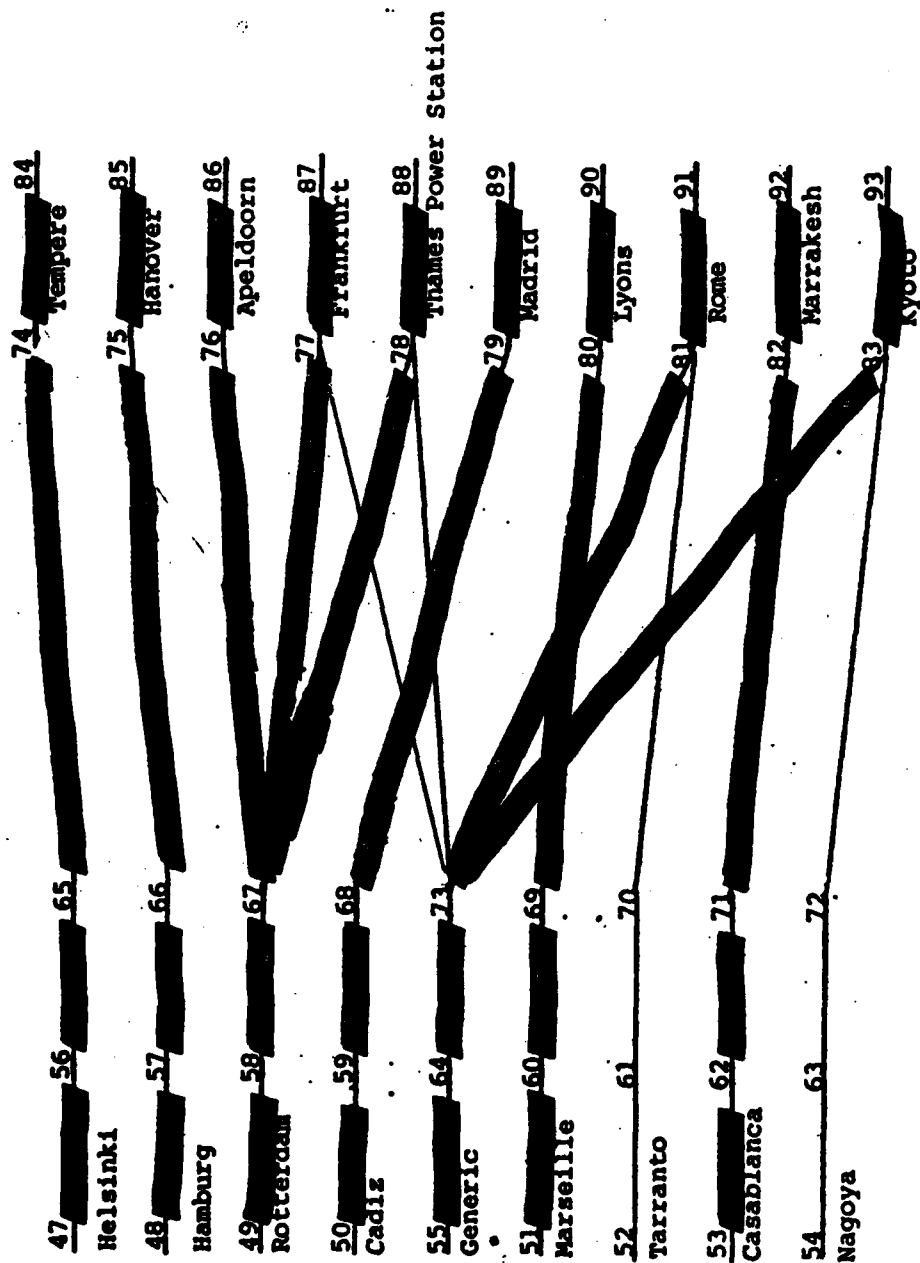
4.4.6C Path Costs Dredging Hampton Roads and Mobile

Export Coal Transportation Network
in the United States



4.4.7A Link Flows Dredging Baltimore and Mobile

Export Coal Transportation Network in Foreign Countries



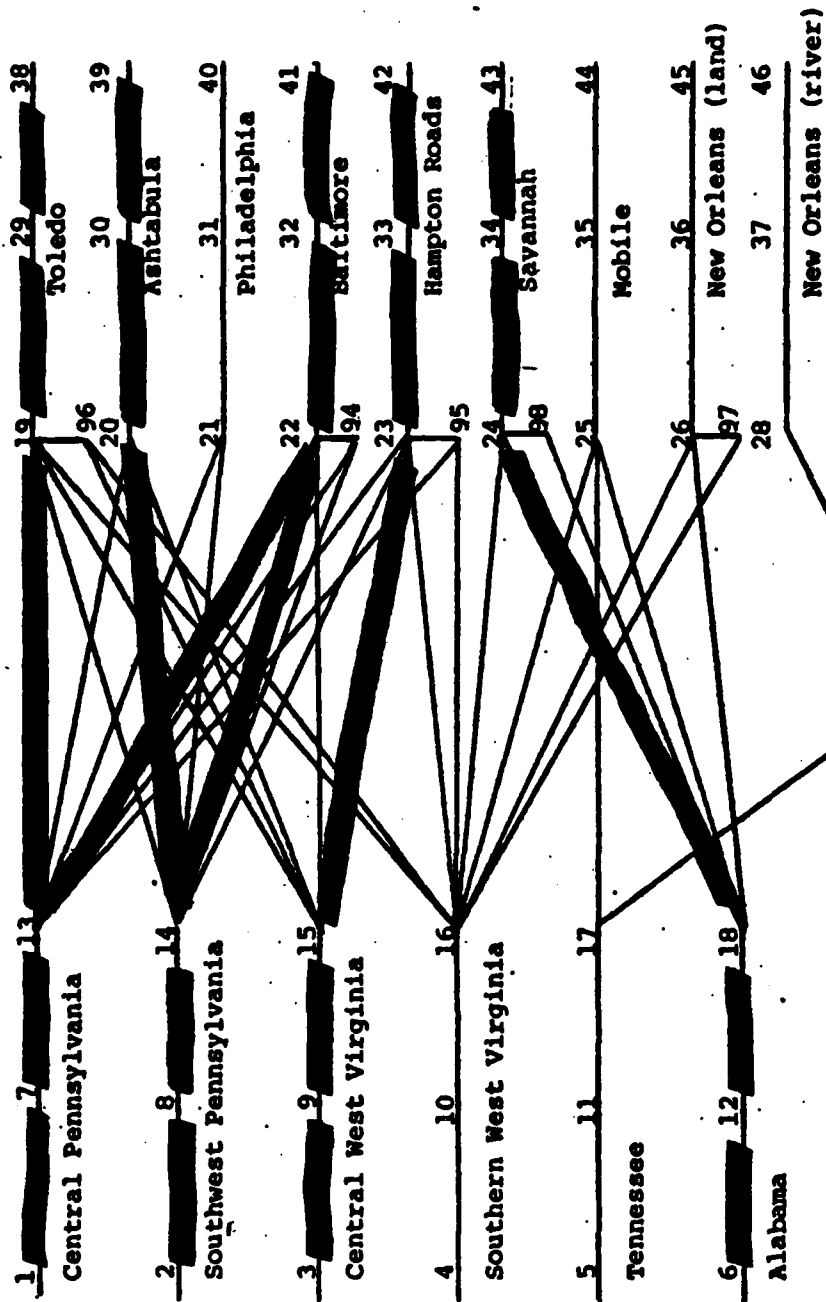
4.4.7B Link Flows Dredging Baltimore and Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	71	60	62	56	56	89	67	62	74	61
2	65	55	67	51	51	83	82	56	68	55
3	73	61	64	58	58	90	69	63	75	62
4	79	67	70	64	64	96	75	69	81	68
5	73	61	64	58	58	91	69	64	76	63
6	68	59	62	56	56	85	68	59	70	58

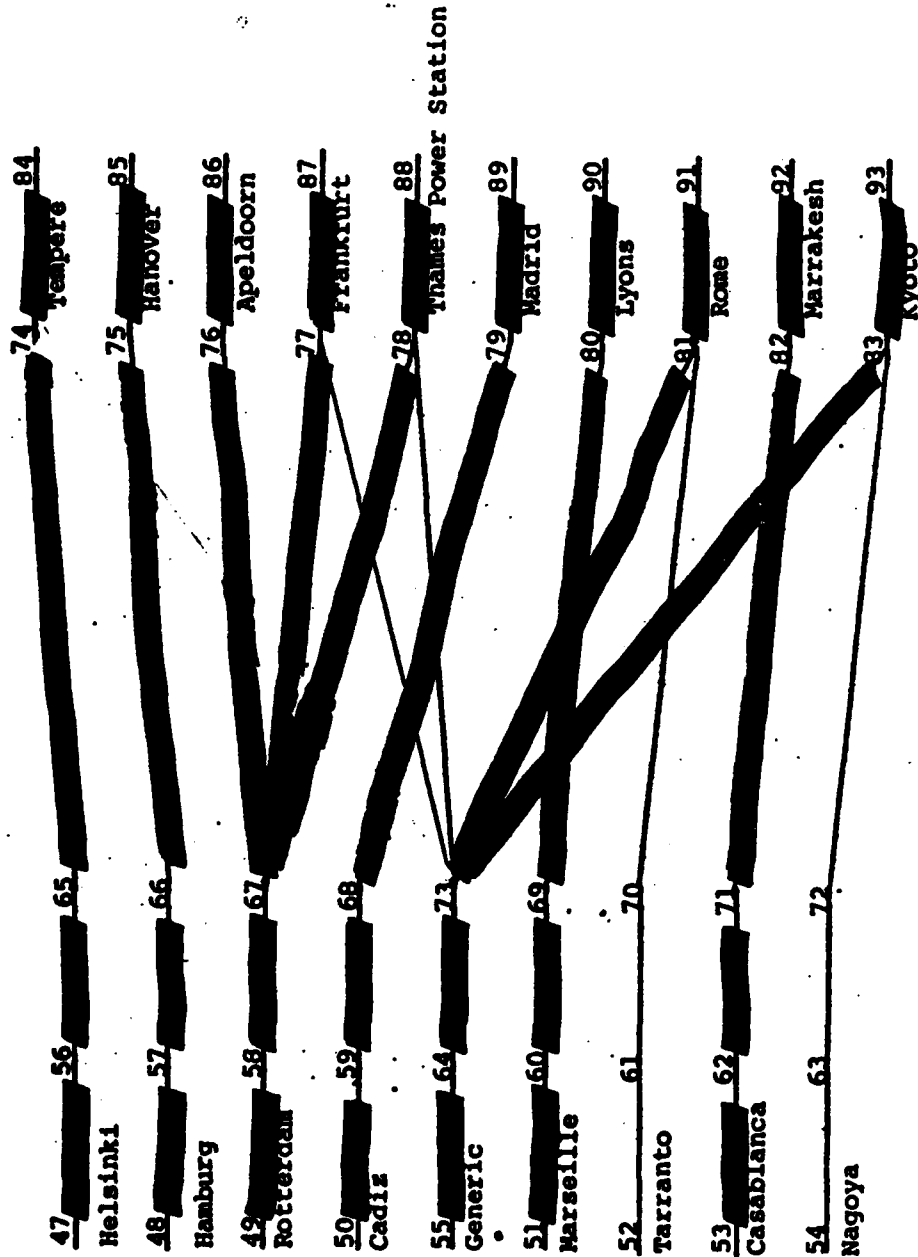
4.4.7C Path Costs Dredging Baltimore and Mobile

**Export Coal Transportation Network
in the United States**



4.4.8A Link Flows Dredging Baltimore,
Hampton Roads and Mobile

Export Coal Transportation Network
in Foreign Countries



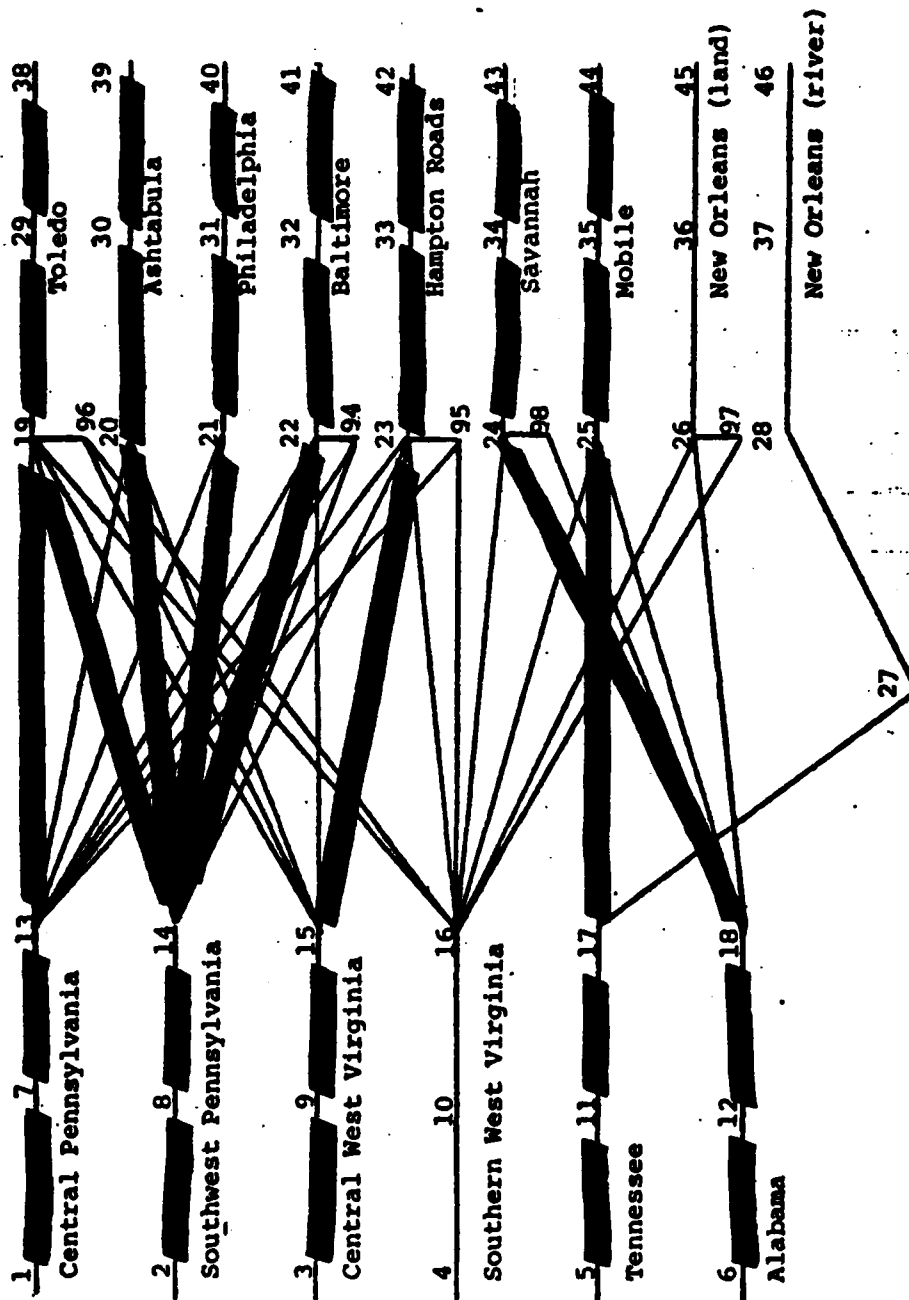
4.4.8B Link Flows Dredging Baltimore,
Hampton Roads and Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	71	60	62	56	56	89	67	62	74	61
2	65	55	67	51	51	83	62	56	68	55
3	73	61	63	57	57	90	68	63	75	62
4	79	67	69	63	63	96	74	69	81	68
5	73	61	64	58	58	91	69	64	76	63
6	68	59	62	56	56	85	68	59	70	58

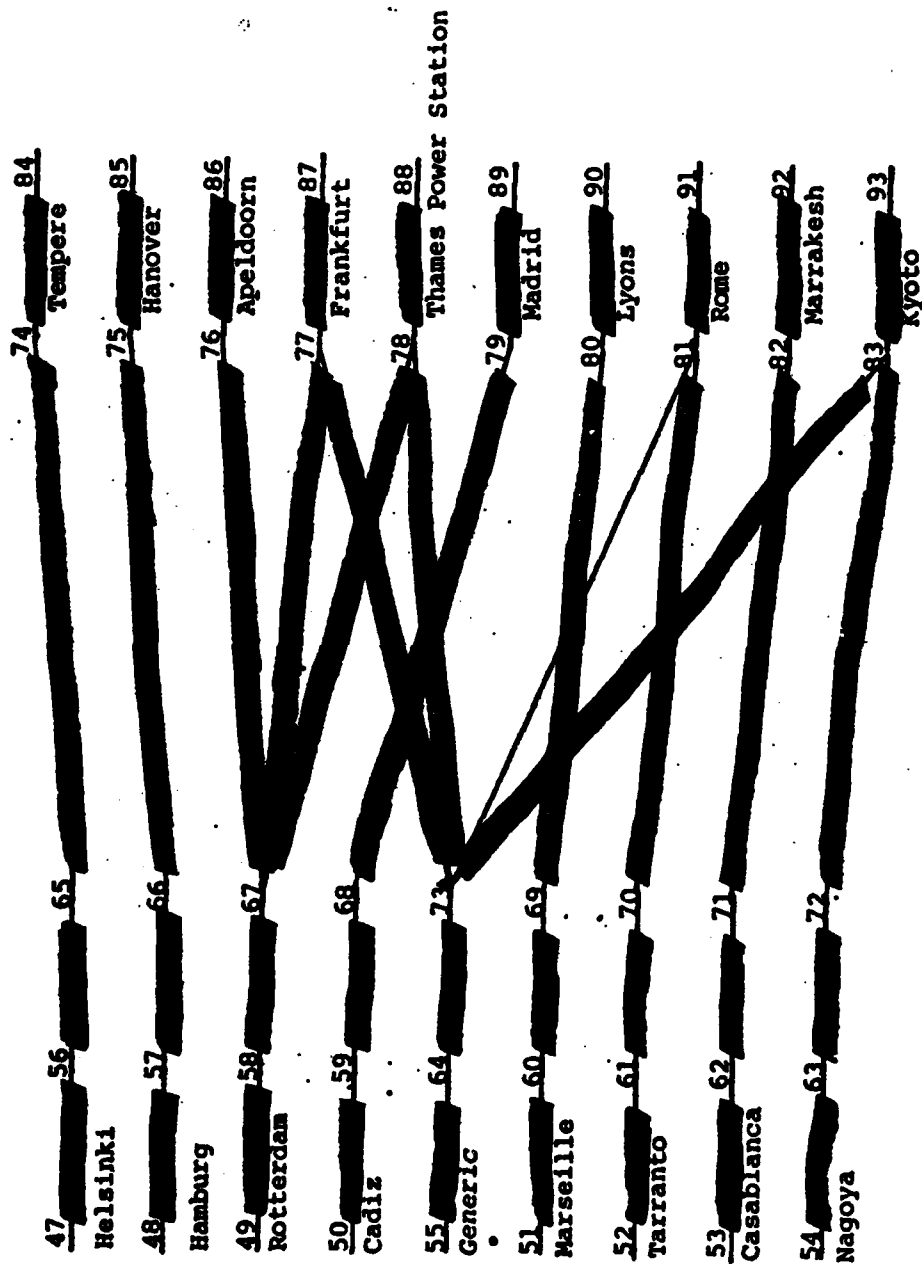
4.4.8C Path Costs Dredging Baltimore, Hampton
Roads and Mobile

Export Coal Transportation Network
in the United States



4.4.9A Link Flows Base Case Series II

Export Coal Transportation Network
in Foreign Countries



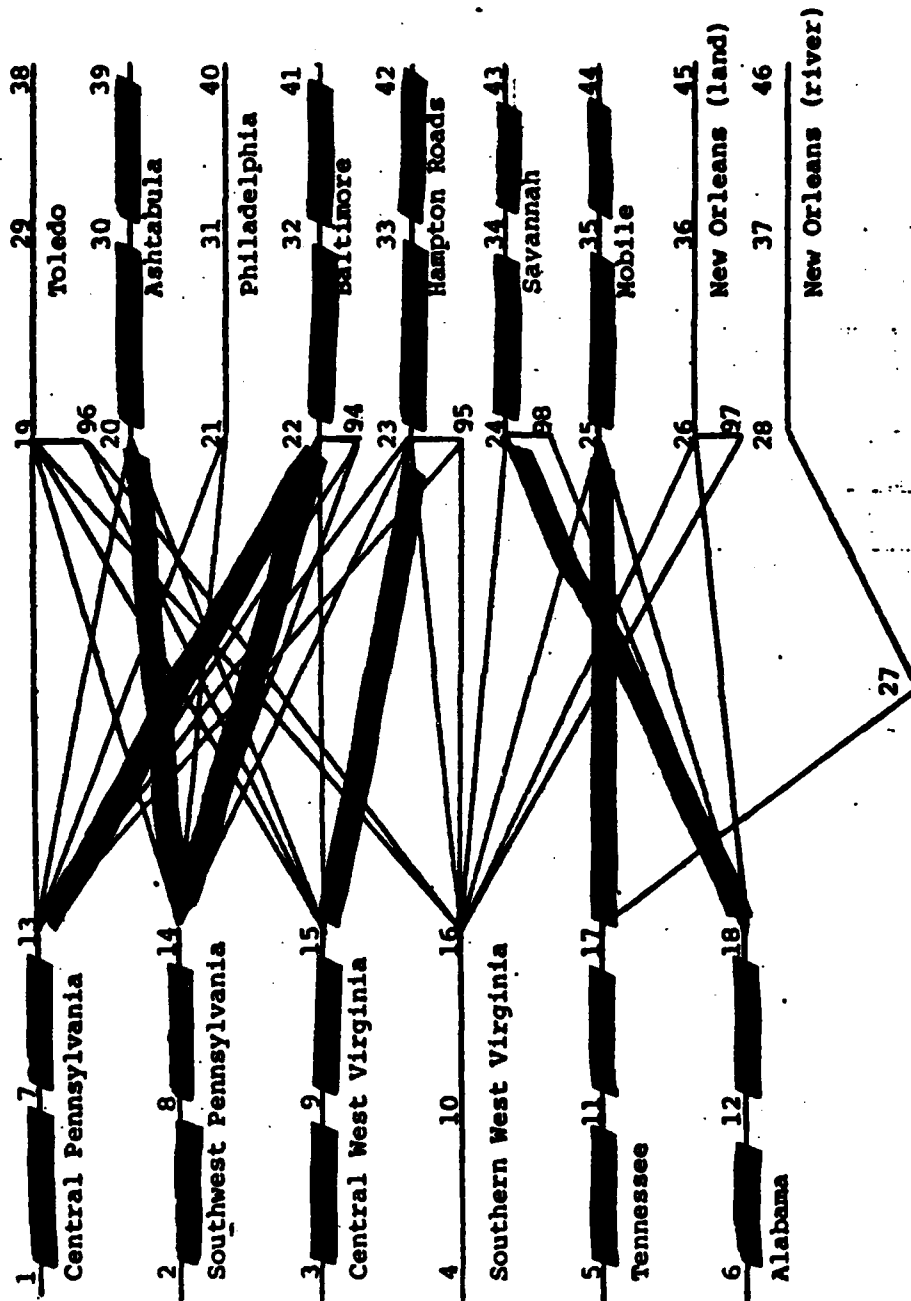
4.4.9B Link Flows Base Case Series II

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	69	73	67	65	94	78	67	79	66
2	63	56	60	54	52	81	65	54	66	53
3	79	67	70	64	64	96	75	69	81	68
4	89	77	80	74	74	106	85	79	91	78
5	76	67	71	65	65	94	76	67	79	66
6	74	65	68	62	62	91	74	65	76	64

4.4.9C Path Costs Base Case Series II

Export Coal Transportation Network in the United States



4.4.10A Link Flows Dredging Baltimore

47	56	74	84
Helsinki		Tempere	
48	57	75	85
Hamburg		Hanover	
49	58	76	86
Rotterdam		Apeldoorn	
50	59	77	87
Cadiz		Frankfurt	
51	60	78	88
Generic		Thames Power	
52	61	79	89
Marseille		Madrid	
53	62	80	90
Tarranto		Lyons	
54	63	81	91
Casablanca		Rome	
55	64	82	92
Nagoya		Marrakesh	
56	65	83	93
		Kyoto	

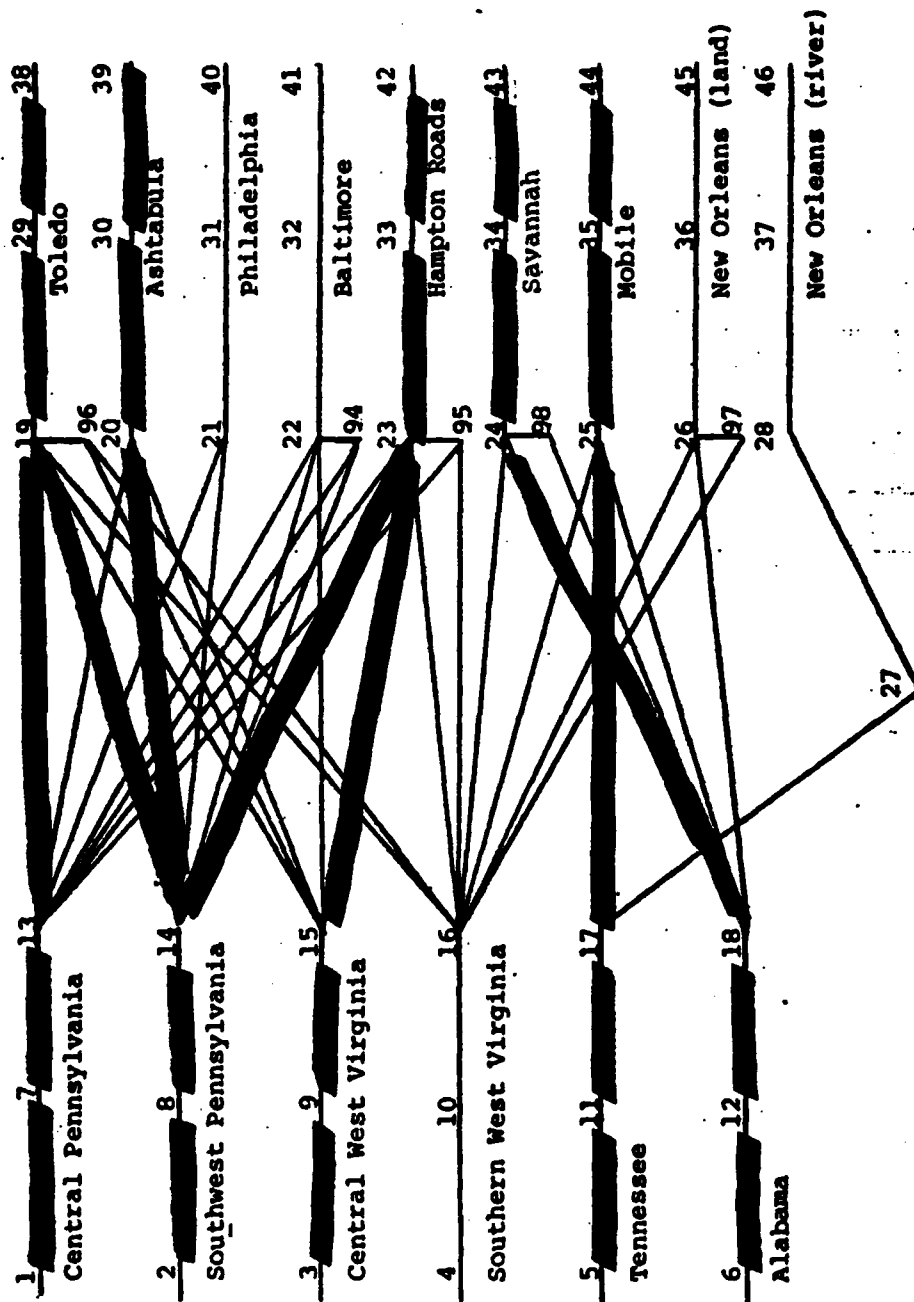
4.4.108 Link Flows Dredging Baltimore

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	67	69	63	63	94	74	67	79	66
2	63	55	57	51	51	81	62	54	66	53
3	79	67	70	64	64	96	75	69	81	68
4	89	77	80	74	74	106	85	79	91	78
5	76	67	71	65	65	94	76	67	79	66
6	74	65	68	62	62	91	74	65	76	64

4.4.10C Path Costs Dredging Baltimore

Export Coal Transportation Network in the United States



4.4.11A Link Flows Dredging Hampton Roads

<u>47</u>	<u>56</u>	<u>65</u>	<u>74</u>	<u>84</u>
Helsinki			Tempere	
<u>48</u>	<u>57</u>	<u>66</u>	<u>75</u>	<u>85</u>
Hamburg			Hanover	
<u>49</u>	<u>58</u>	<u>67</u>	<u>76</u>	<u>86</u>
Rotterdam			Apeidoorn	
<u>50</u>	<u>59</u>	<u>68</u>	<u>77</u>	<u>87</u>
Cadiz			Frankfurt	
<u>55</u>	<u>64</u>	<u>73</u>	<u>78</u>	<u>88</u>
Generic			Thames Power Station	
<u>51</u>	<u>60</u>	<u>69</u>	<u>79</u>	<u>89</u>
Marseille			Madrid	
<u>52</u>	<u>61</u>	<u>70</u>	<u>80</u>	<u>90</u>
Tarranto			Lyons	
<u>53</u>	<u>62</u>	<u>71</u>	<u>81</u>	<u>91</u>
Casablanca			Rome	
<u>54</u>	<u>63</u>	<u>72</u>	<u>82</u>	<u>92</u>
Nagoya			Marrakesh	
			<u>83</u>	<u>93</u>
			Kyoto	

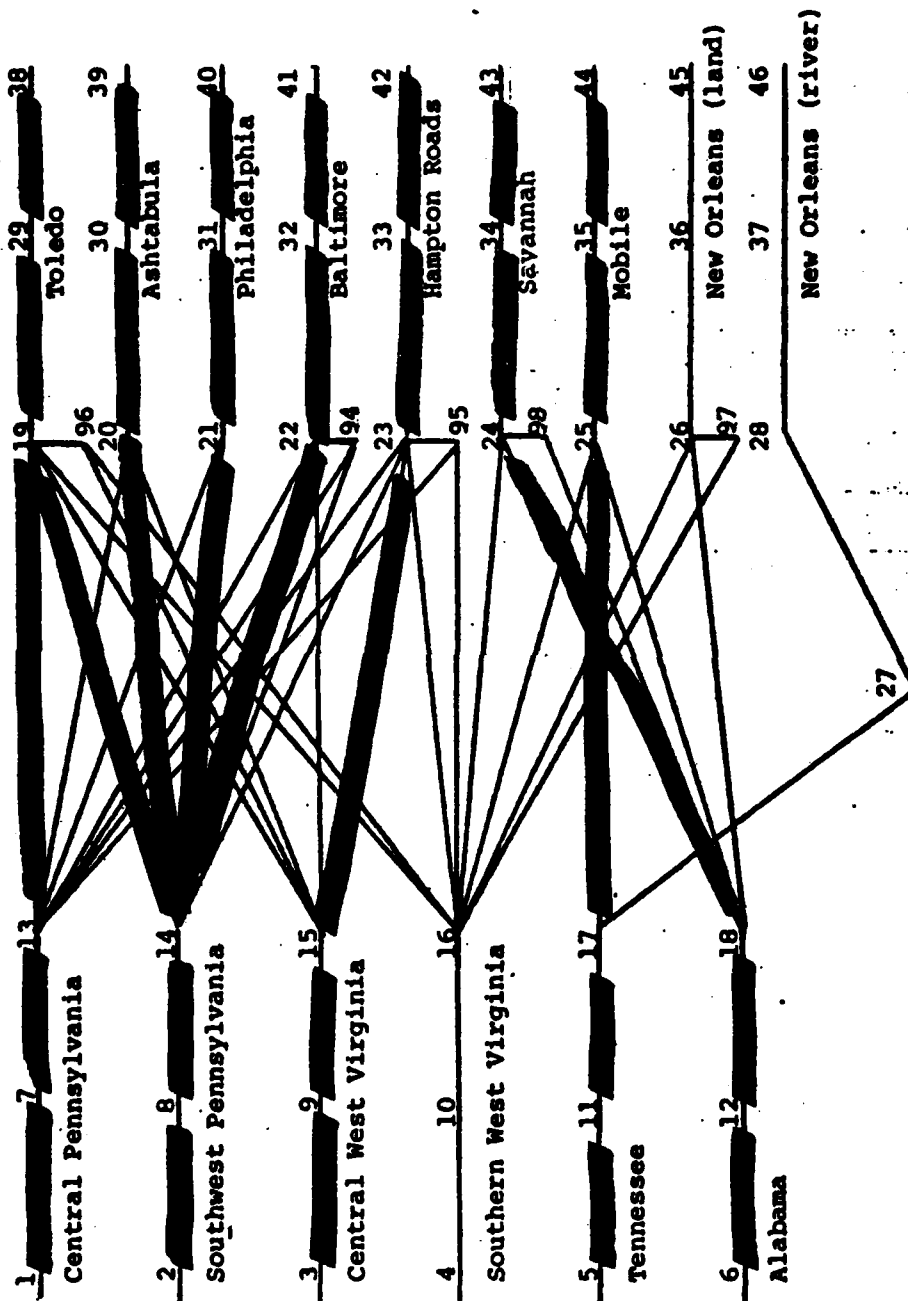
4.4.11B Link Flows Dredging Hampton Roads

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	69	72	66	65	94	77	67	79	66
2	63	56	60	54	52	81	65	54	66	53
3	79	67	69	63	63	96	74	69	81	68
4	89	77	79	73	73	106	84	79	91	78
5	76	67	71	65	65	94	76	67	79	66
6	74	65	68	62	62	91	74	65	76	64

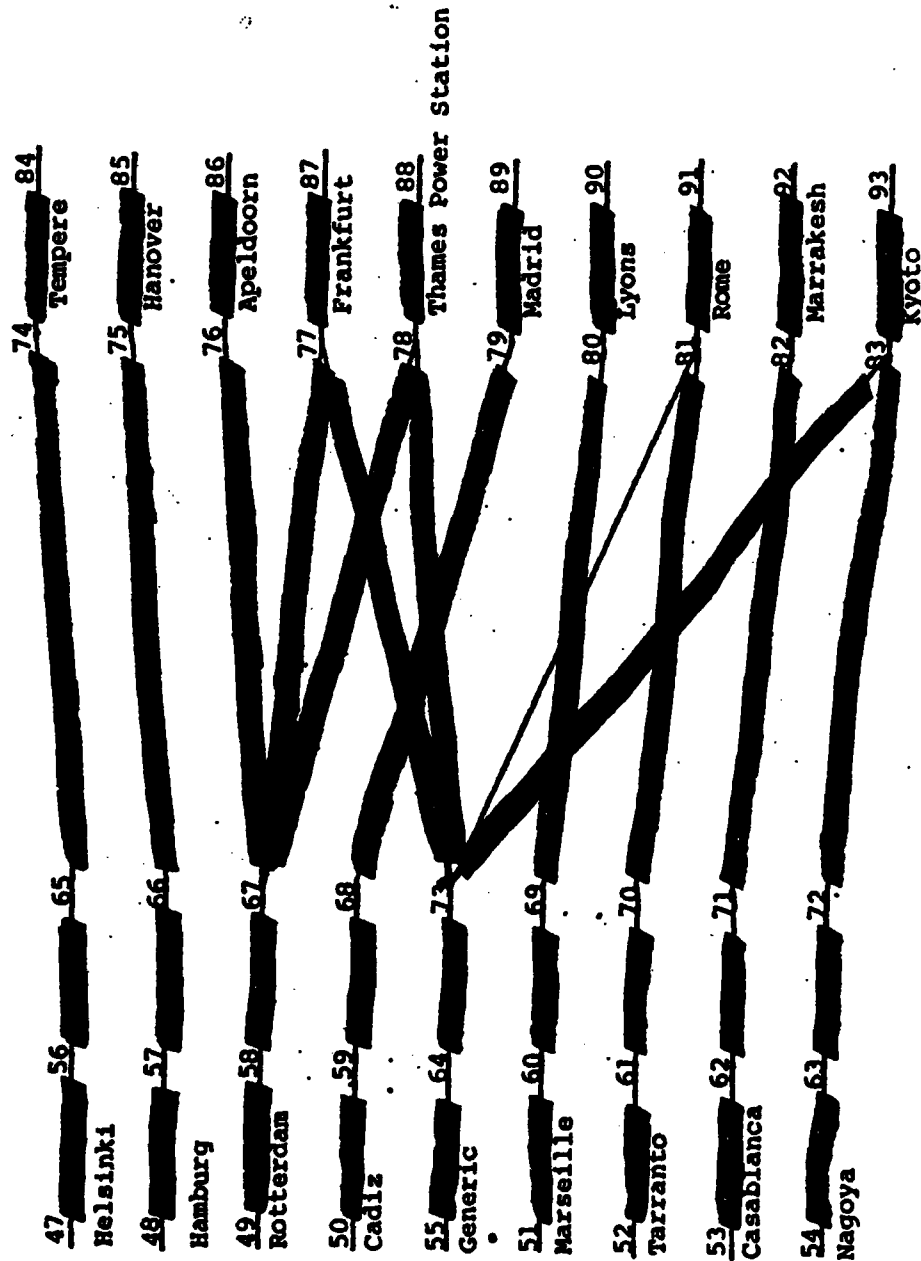
4.4.11C Path Costs Dredging Hampton Roads

Export Coal Transportation Network
in the United States



4.4.12A Link Flows Dredging Mobile

Export Coal Transportation Network
in Foreign Countries



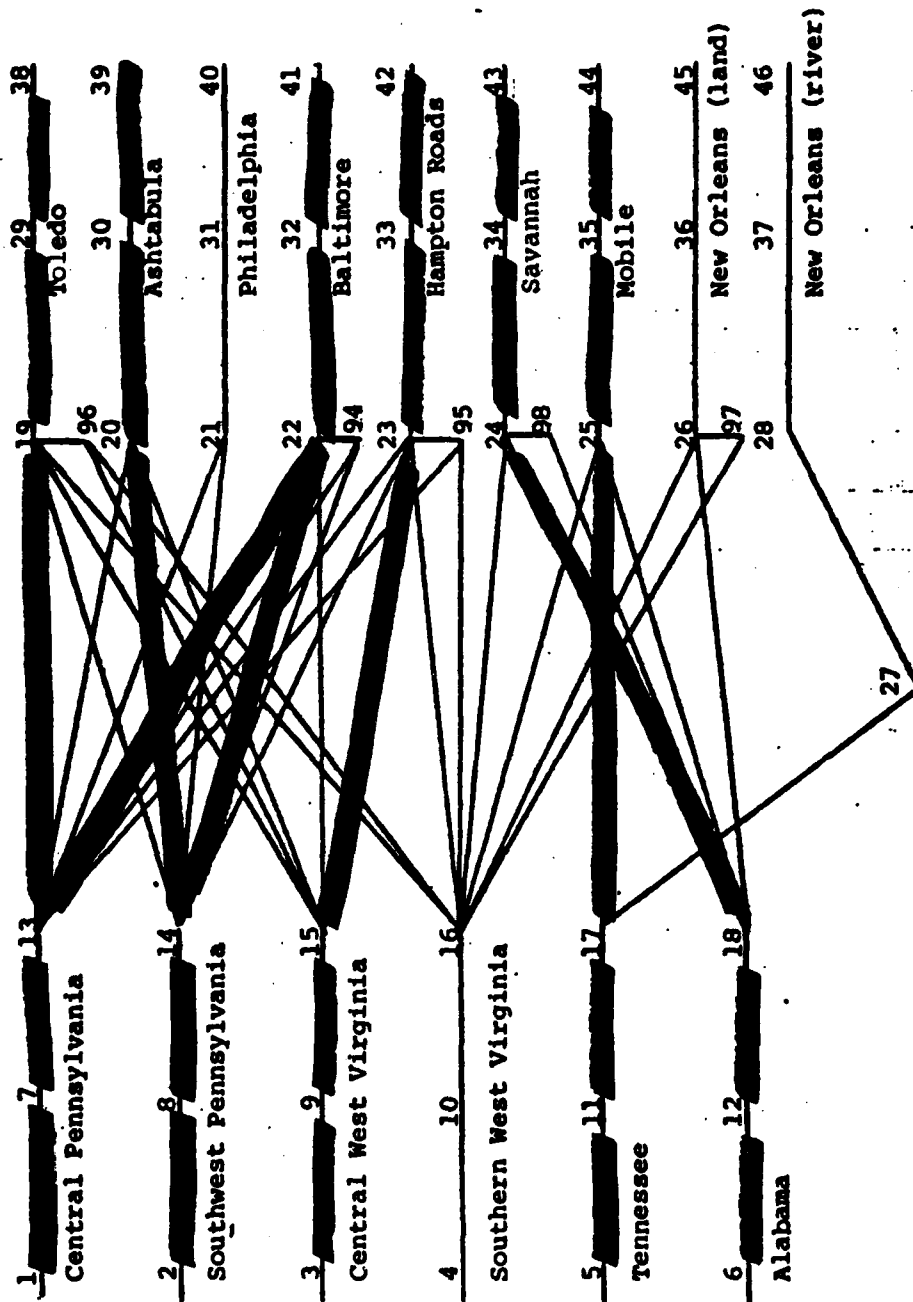
4.4.12B Link Flows Dredging Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	69	73	67	65	94	78	67	79	66
2	63	56	60	54	52	81	65	54	66	53
3	79	67	70	64	64	96	75	69	81	68
4	89	77	80	74	74	106	85	79	91	78
5	76	64	67	61	61	94	72	67	79	66
6	74	65	68	62	62	91	74	65	76	64

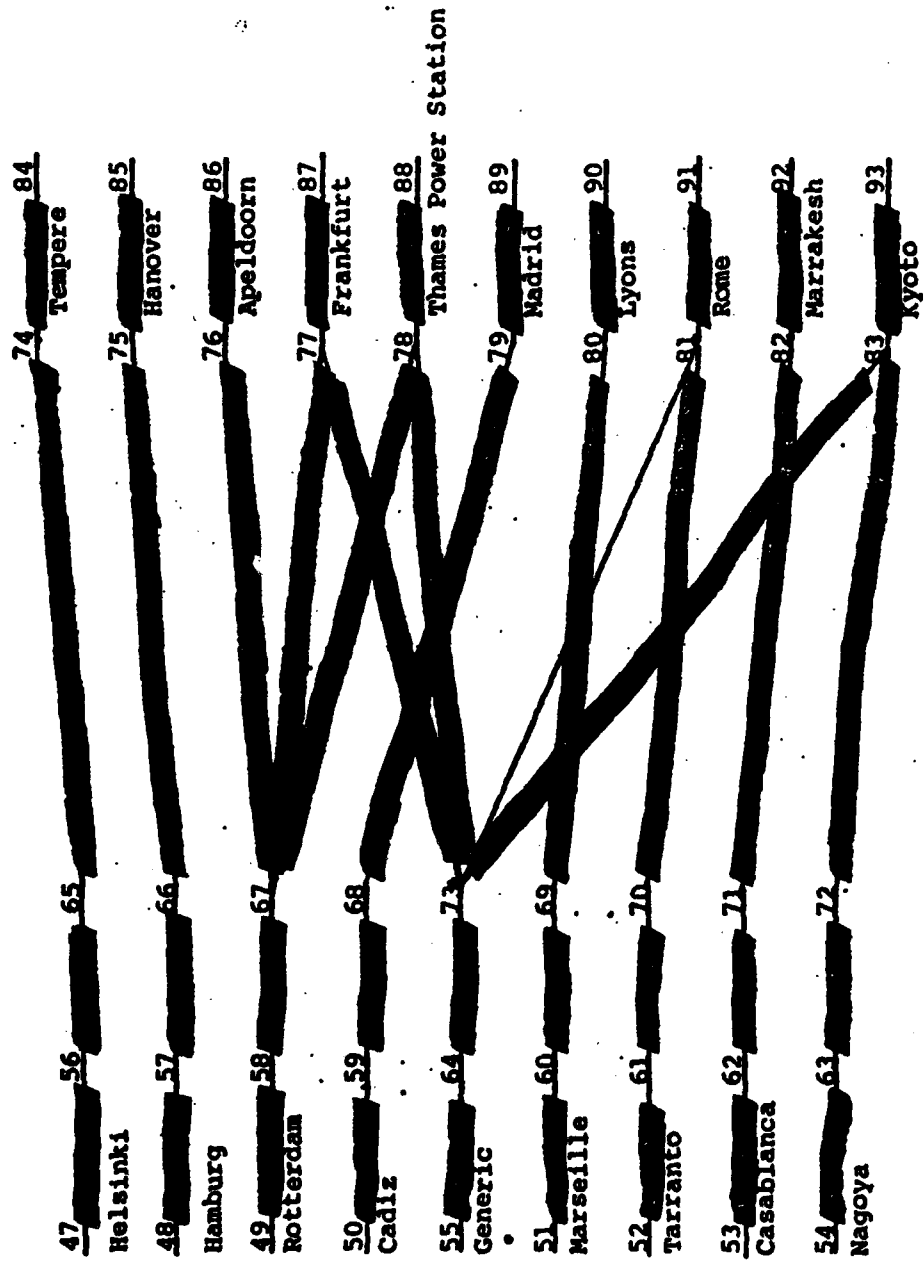
4.4.12C Path Costs Dredging Mobile

Export Coal Transportation Network in the United States



4.4.13A Link Flows Dredging Baltimore and Hampton Roads

Export Coal Transportation Network
in Foreign Countries



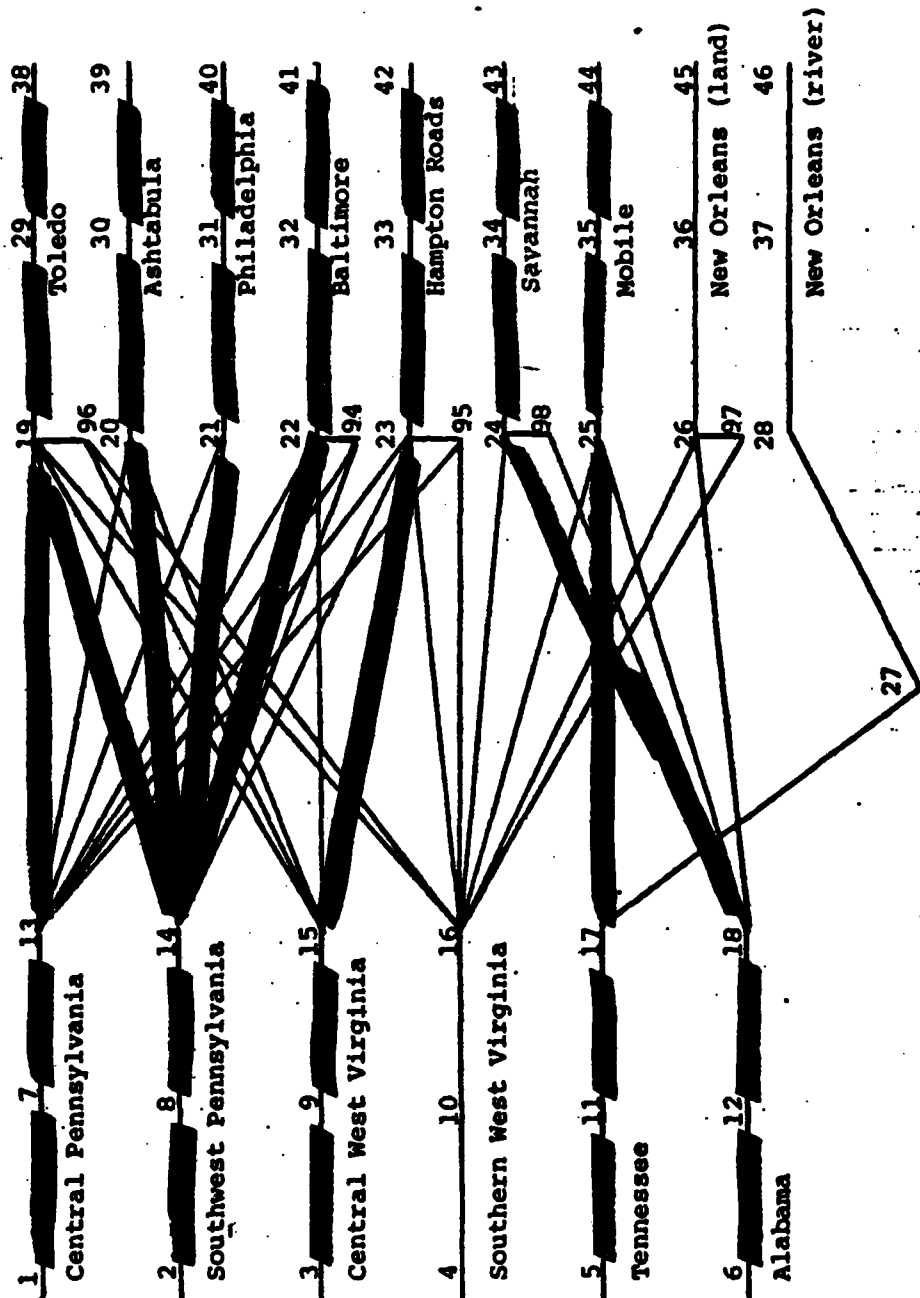
4.4.13B Link Flows Dredging Baltimore and Hampton Roads

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	67	69	63	63	94	74	67	79	66
2	63	55	57	51	51	81	62	54	66	53
3	79	67	69	63	63	96	74	69	81	68
4	89	77	79	73	73	106	84	79	91	78
5	76	67	71	65	65	94	76	67	79	66
6	74	65	68	62	62	91	74	65	76	64

4.4.13C Path Costs Dredging Baltimore and Hampton Roads

Export Coal Transportation Network
in the United States



4.4.14A Link Flows Dredging Hampton Roads and Mobile

189

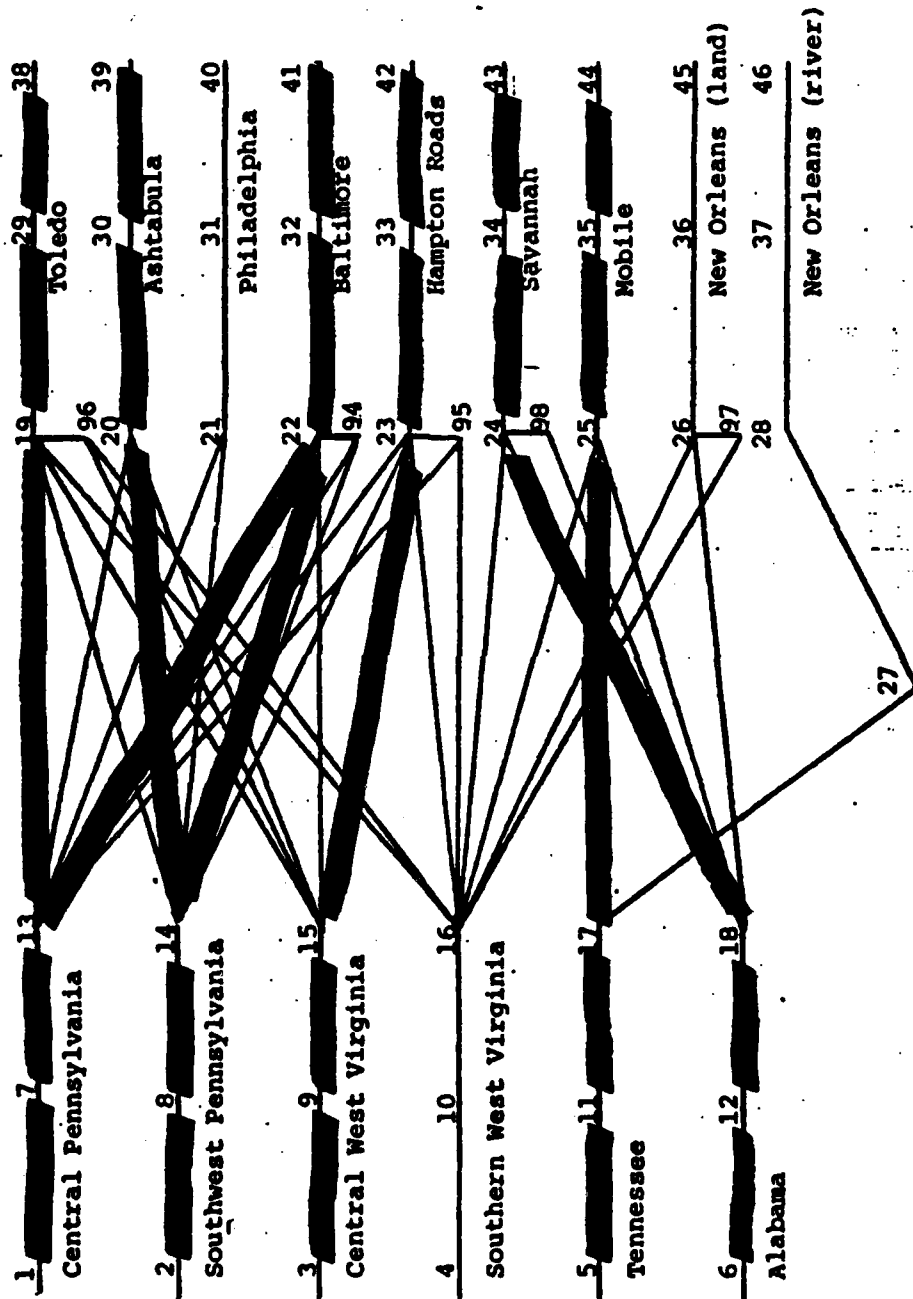
4.4.14B Link Flows Dredging Hampton Roads and Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	69	72	66	65	94	77	67	79	66
2	63	56	60	63	63	96	74	69	66	53
3	79	67	69	63	63	96	74	69	81	68
4	89	77	79	73	73	106	84	72	91	78
5	76	64	67	61	61	94	72	67	79	66
6	74	64	68	62	62	91	74	65	76	64

4.4.14C Path Costs Dredging Hampton Roads and Mobile

Export Coal Transportation Network in the United States



4.4.15A Link Flows Dredging Baltimore and Mobile

AD-A146 013

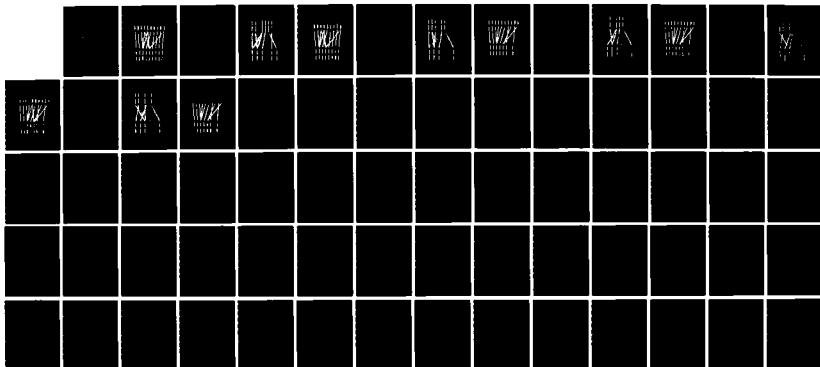
ANALYSIS OF THE TRANSPORTATION NETWORK FOR THE EXPORT
OF US STEAM COAL(U) MASSACHUSETTS INST OF TECH
CAMBRIDGE S R LINDBERG SEP 84

3/3

UNCLASSIFIED

F/G 5/3

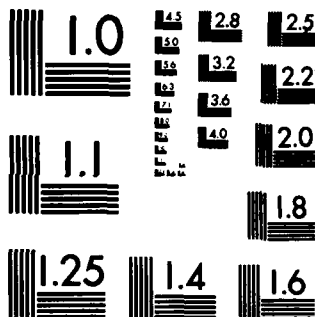
NL



END

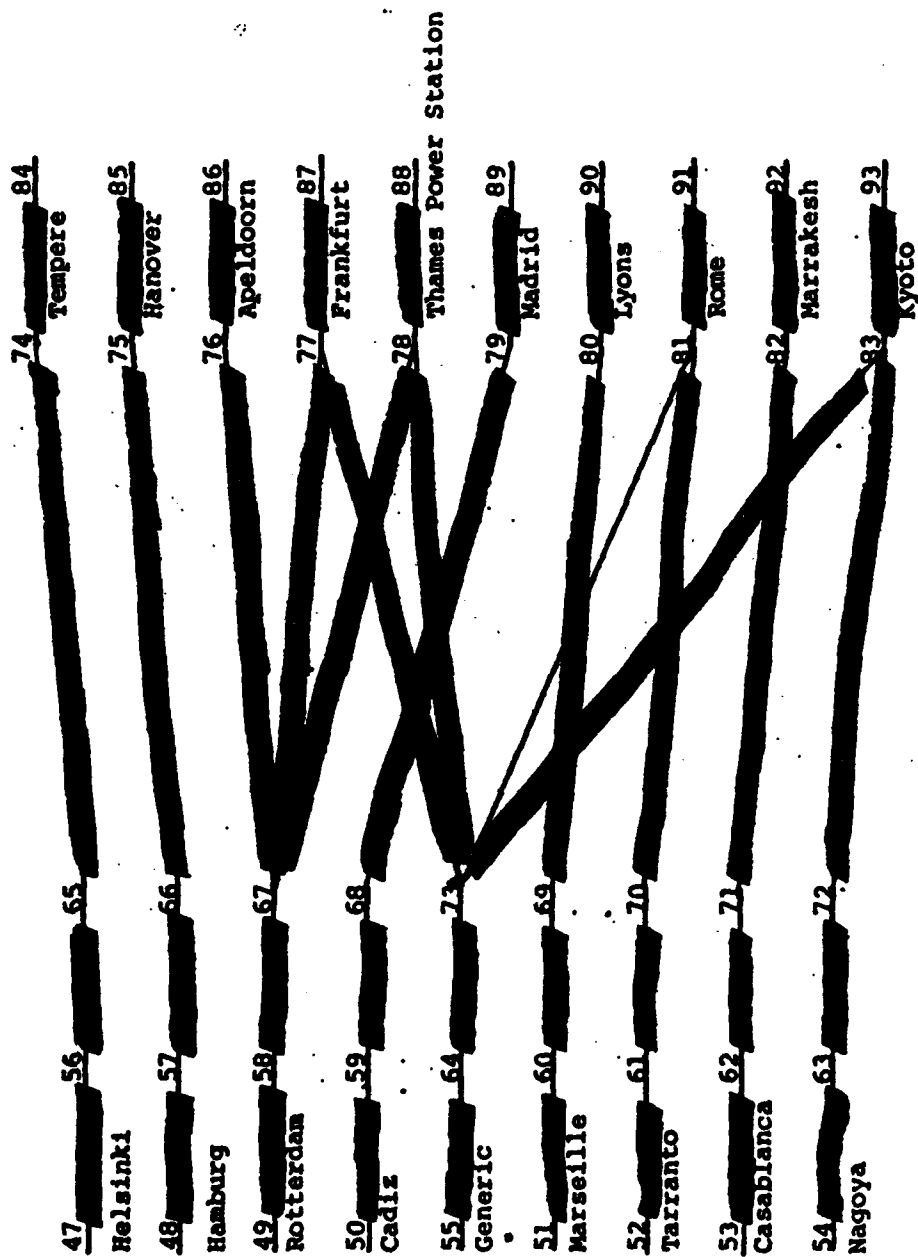
FILED

DNIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Export Coal Transportation Network
in Foreign Countries



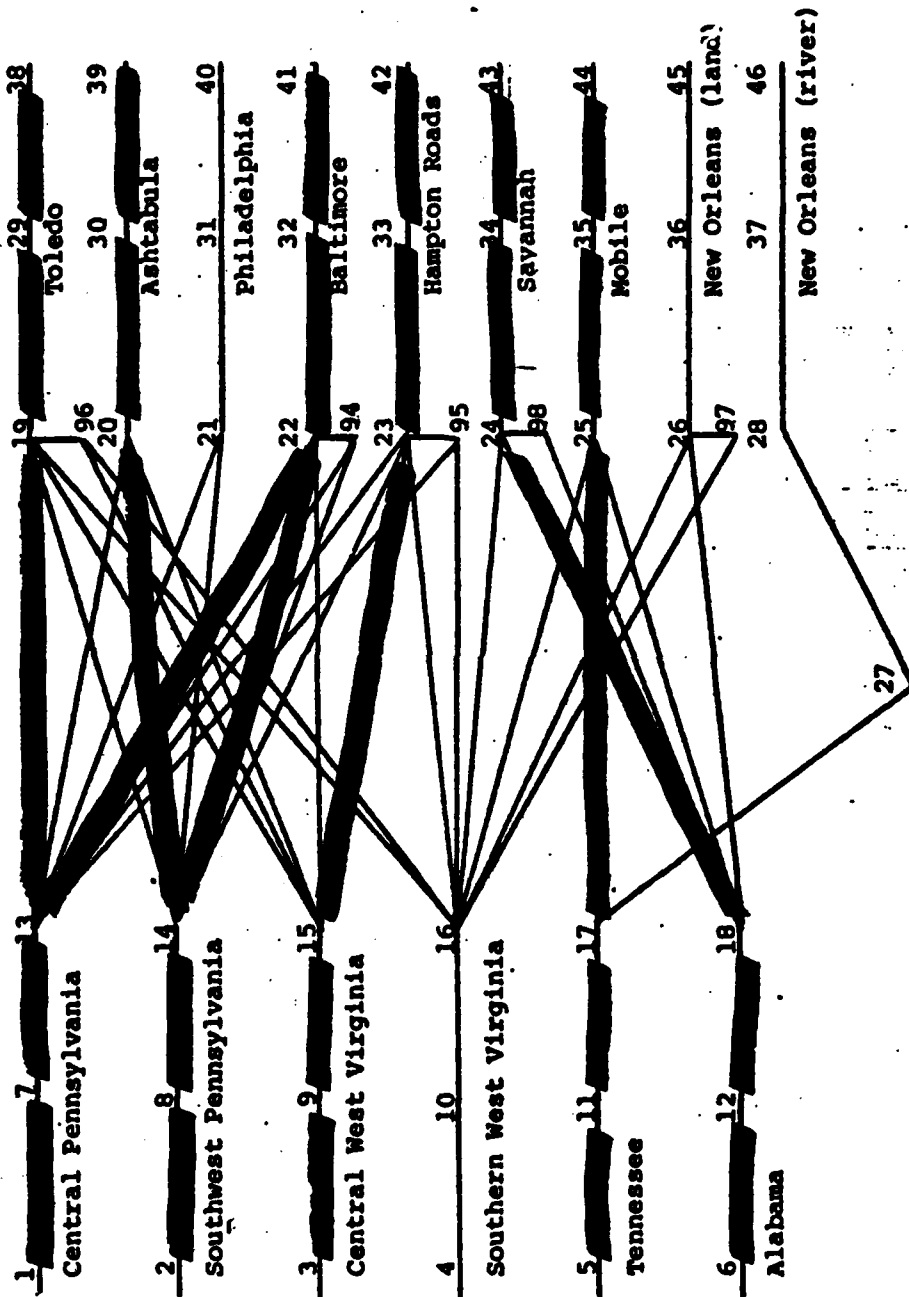
4.4.15B Link Flows Dredging Baltimore and Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	67	69	63	63	94	74	67	79	66
2	63	55	57	51	51	81	62	54	66	53
3	79	67	70	64	64	96	75	69	81	68
4	89	77	80	74	74	106	85	79	91	78
5	76	64	67	61	61	94	72	67	79	66
6	74	65	68	62	62	91	74	65	76	64

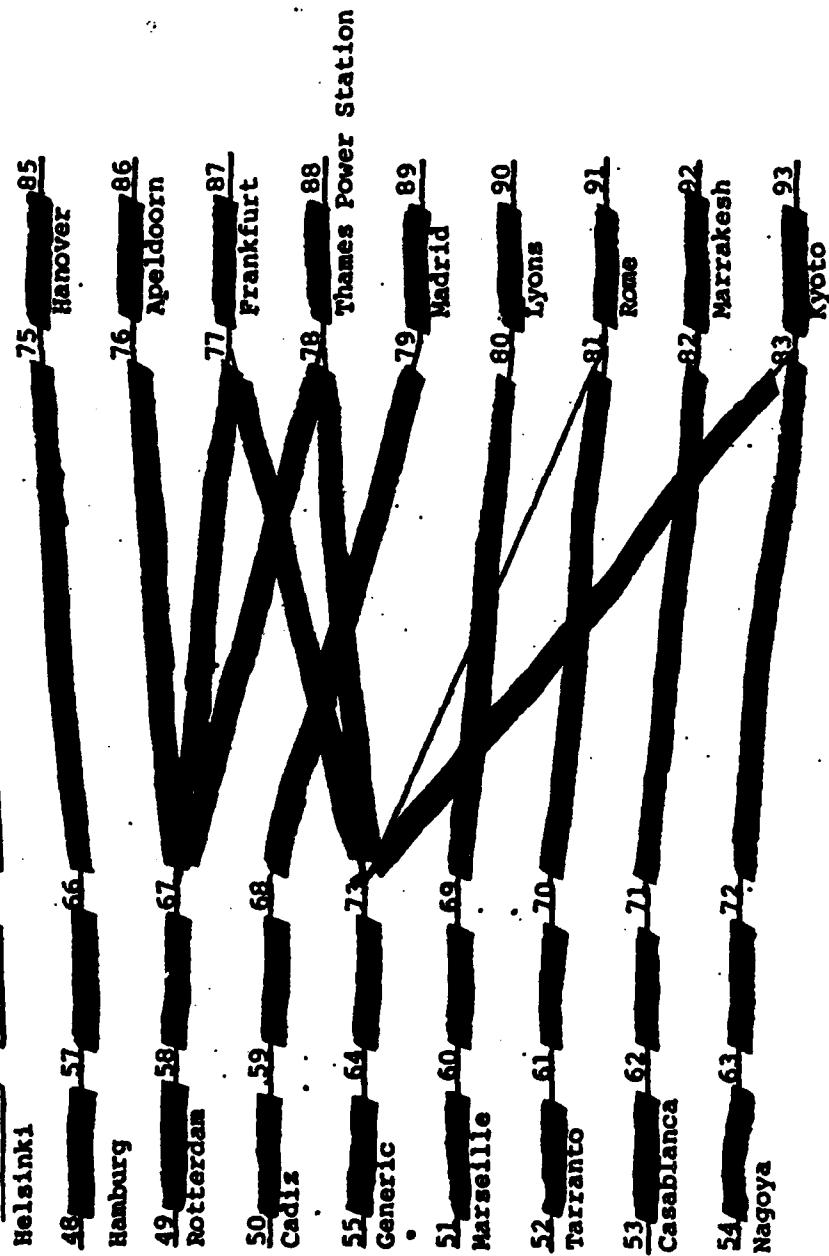
4.4.15C Path Costs Dredging Baltimore and Mobile

Export Coal Transportation Network in the United States



4.4.16A Link Flows Dredging Baltimore,
Hampton Roads and Mobile

Export Coal Transportation Network



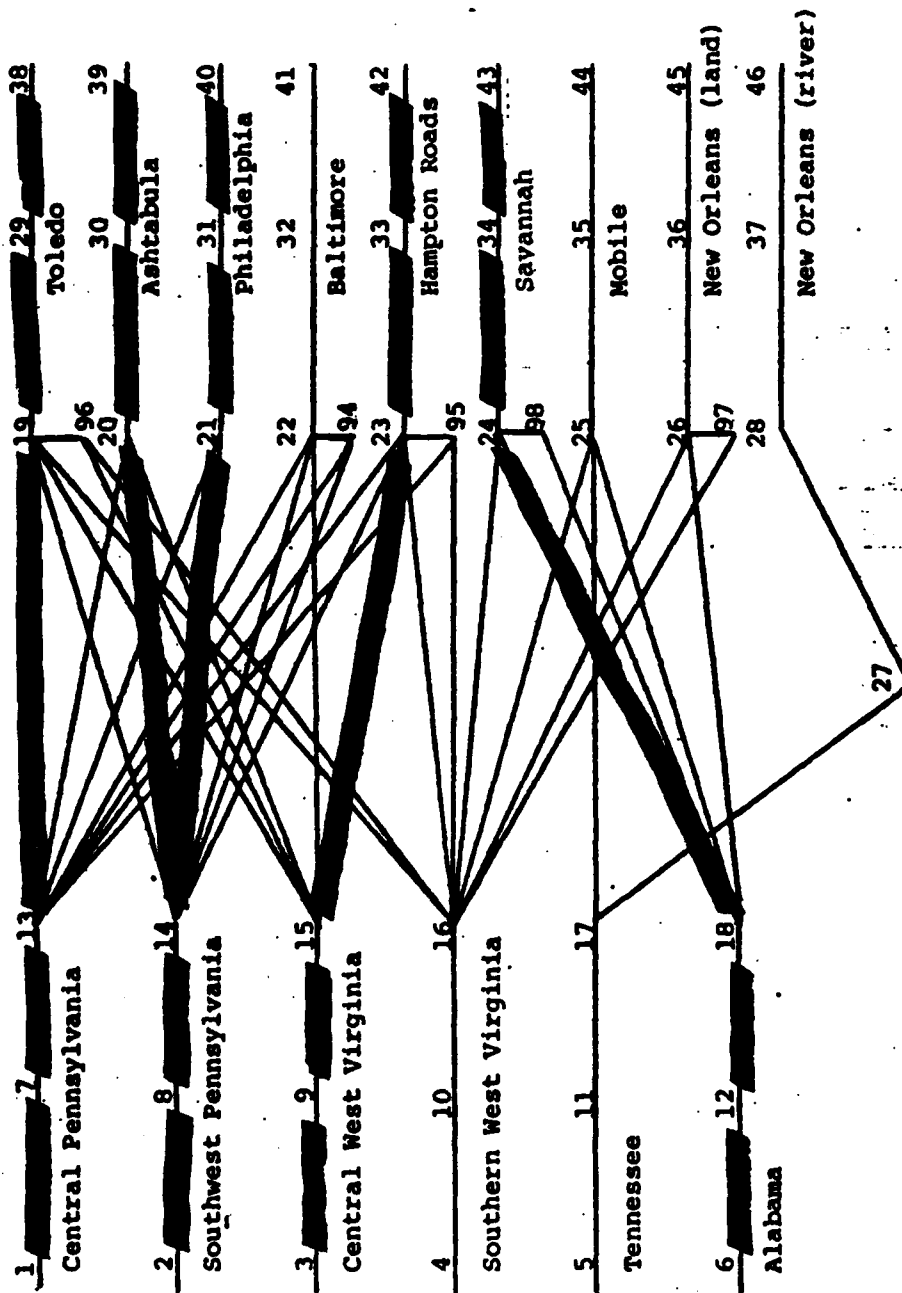
4.4.16B Link Flows Dredging Baltimore,
Hampton Roads and Mobile

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	76	67	69	63	63	94	74	67	79	66
2	63	55	57	51	51	81	62	54	66	53
3	79	67	69	63	63	96	74	69	81	68
4	89	77	79	73	73	106	84	79	91	78
5	76	64	67	61	61	94	72	67	79	66
6	74	65	86	62	62	91	74	65	76	64

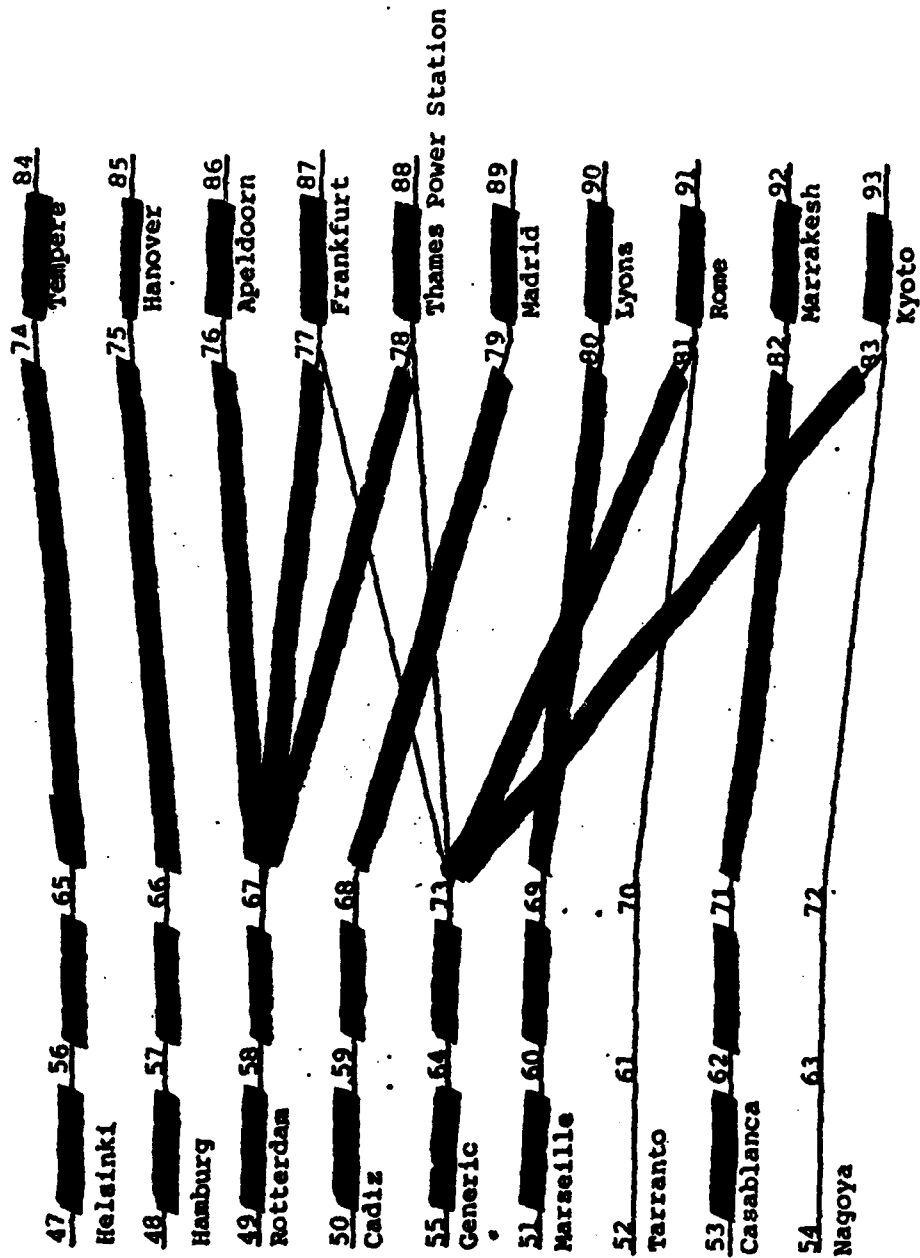
4.4.16C Path Costs Dredging Baltimore, Hampton
Roads and Mobile

Export Coal Transportation Network in the United States



4.17A Link Flows Sulfur Penalty Base Case

Export Coal Transportation Network
in Foreign Countries



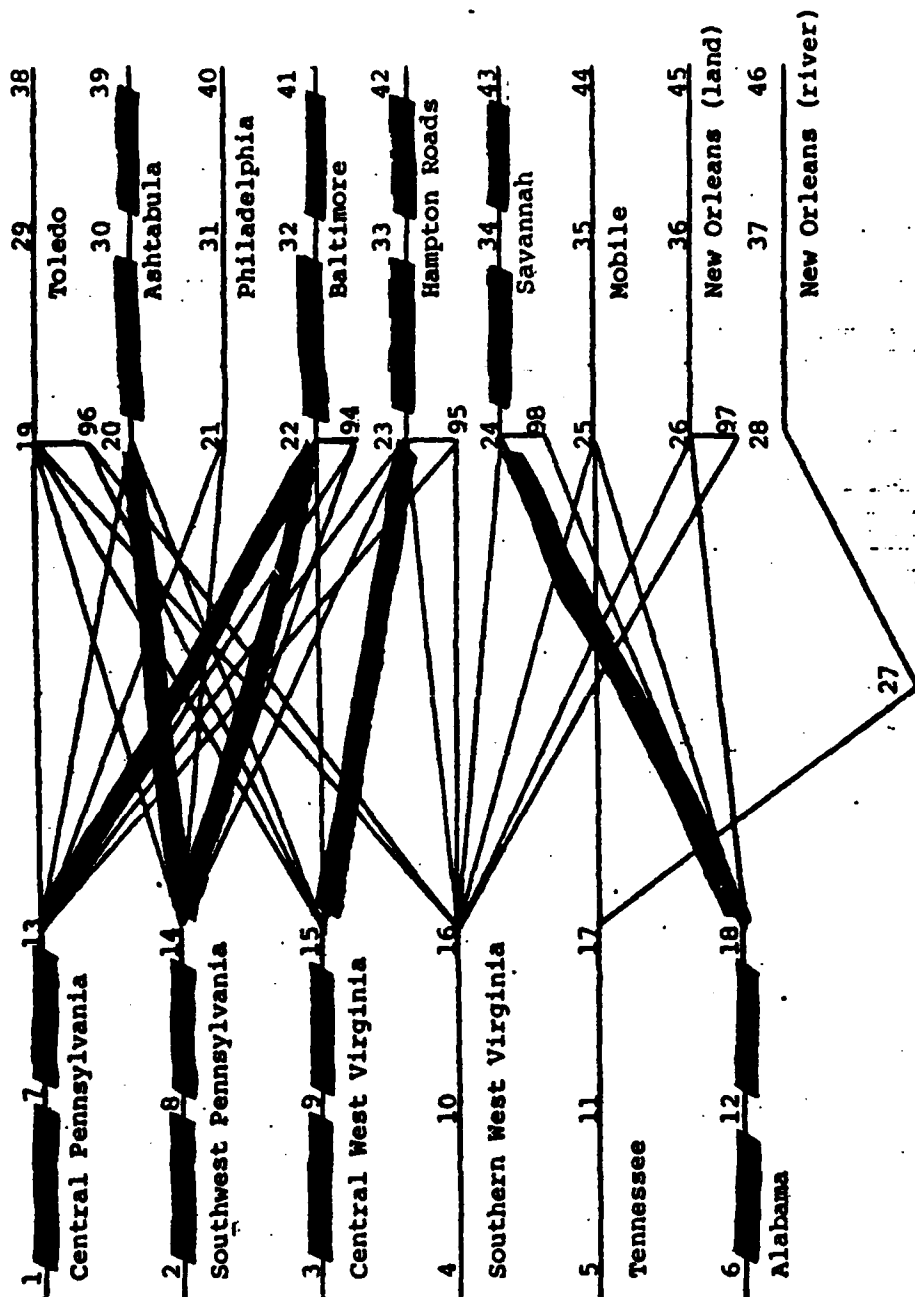
4.17B Link Flows Sulfur Penalty Base Case

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	69	62	66	60	58	87	71	60	72	59
2	67	60	64	58	56	85	69	58	70	57
3	73	61	64	58	58	90	69	63	75	62
4	79	67	70	64	64	96	75	69	81	68
5	77	68	72	66	66	95	77	68	80	67
6	72	63	66	60	60	89	72	63	74	62

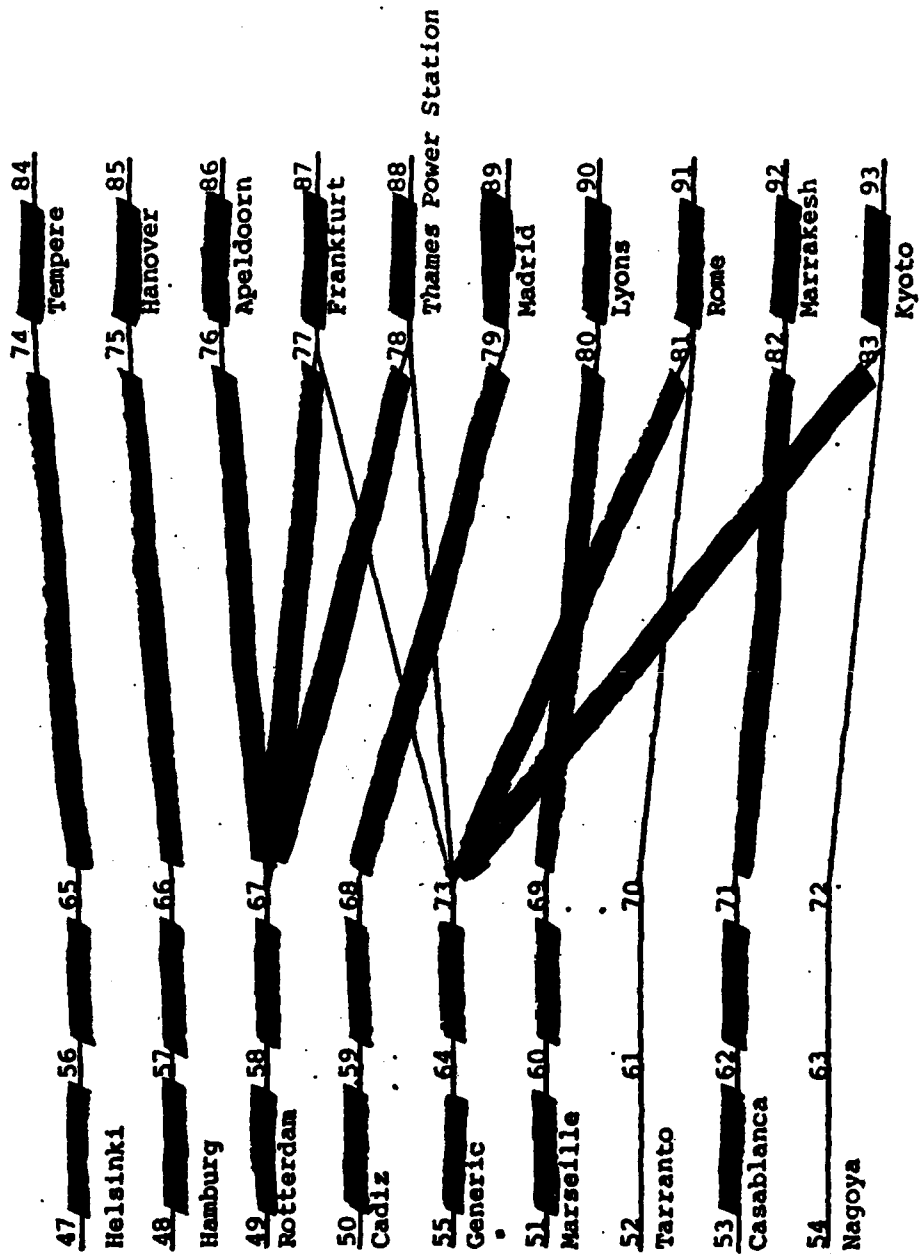
4.17C Path Costs Sulfur Penalty Base Case

Export Coal Transportation Network in the United States



4.18A Link Flows Sulfur Penalty Dredging Baltimore

Export Coal Transportation Network
in Foreign Countries

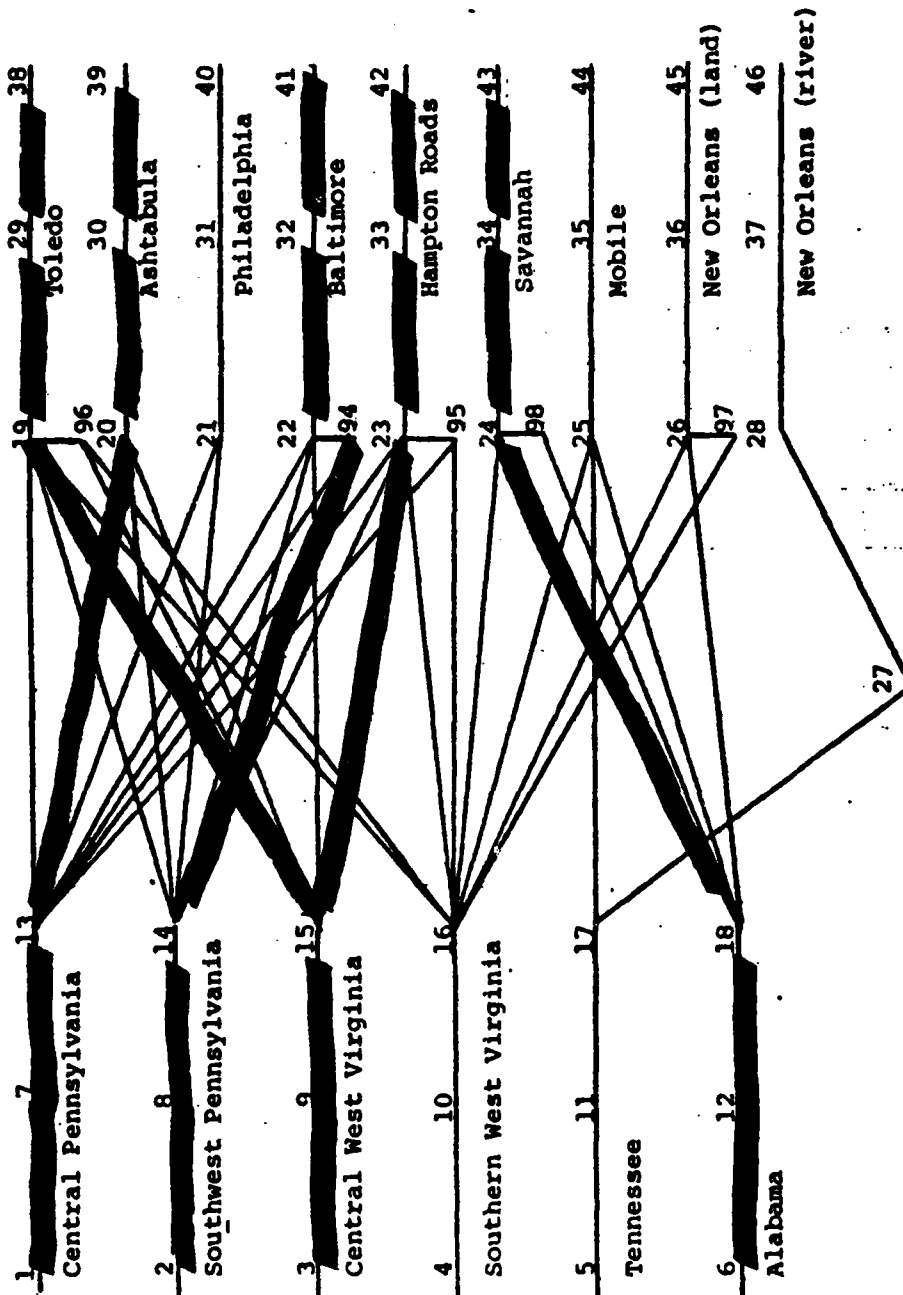


ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	71	60	62	56	56	89	67	62	74	61
2	69	59	61	55	55	87	66	60	72	59
3	73	61	64	58	58	90	69	63	75	62
4	79	67	79	64	64	96	75	69	81	68
5	77	68	72	66	66	95	77	68	80	67
6	72	63	66	60	60	89	72	63	74	62

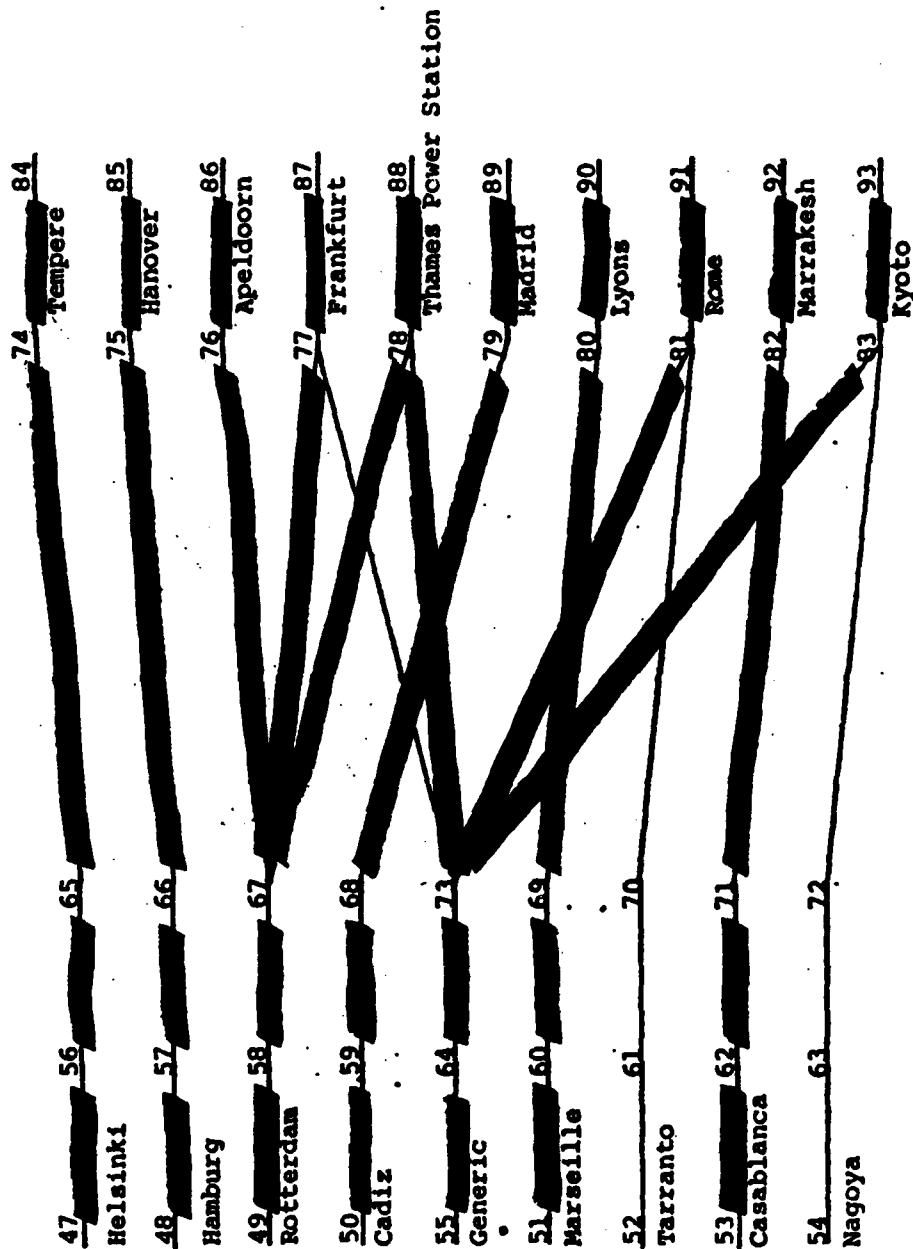
4.18C Path Costs Sulfur Penalty Dredging Baltimore

Export Coal Transportation Network in the United States



4.19A Link Flows New Vessel & Rail Costs Base Case

Export Coal Transportation Network
in Foreign Countries



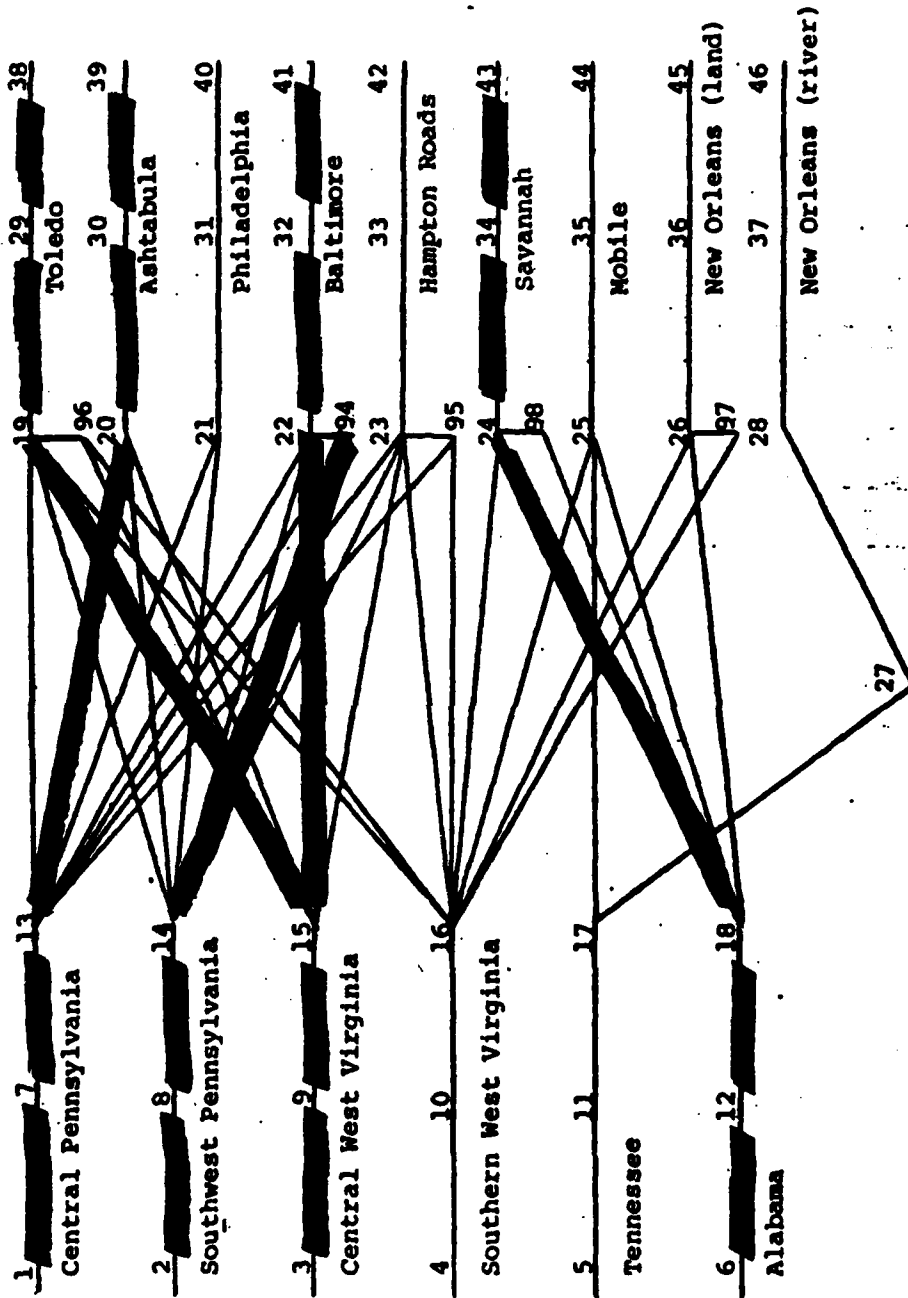
4.19B Link Flows New Vessel & Rail Costs Base Case

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	79	72	76	70	68	97	81	70	82	69
2	78	71	75	69	67	96	80	69	81	68
3	80	72	76	70	69	98	81	71	83	70
4	89	81	85	79	78	107	90	80	92	79
5	86	74	78	72	72	104	83	77	90	76
6	78	68	72	66	66	97	77	70	82	69

4.19C Path Costs New Vessel & Rail Costs Base Case

Export Coal Transportation Network in the United States



4.20A Link Flows New Vessel & Rail Costs Dredging Baltimore

47	56	65	74	84
Helsinki			Tempere	
48	57	66	75	85
Hamburg			Hanover	
49	58	67	76	86
Rotterdam			Apeldoorn	
50	59	68	77	87
Cadiz			Frankfurt	
55	64	73	78	88
Generic			Thames Power Station	
51	60	69	79	89
Marseille			Madrid	
52	61	70	80	90
Tarranto			Lyons	
53	62	71	81	91
Casablanca			Rome	
54	63	72	82	92
Nagoya			Marrakesh	
			83	93
			Kyoto	

4.20B Link Flows New Vessel & Rail Costs Dredging Baltimore

ORIGIN TO DESTINATION COST MATRIX

UTILITIES	84	85	86	87	88	89	90	91	92	93
MINES										
1	79	72	76	70	68	97	81	70	82	69
2	78	70	73	67	67	96	78	69	81	68
3	80	71	74	68	68	98	79	71	83	70
4	89	81	85	79	78	107	90	80	92	79
5	86	74	78	72	72	104	83	77	90	76
6	78	68	72	66	66	97	77	70	82	69

4.20C Path Costs New Vessel & Rail Costs Dredging
Baltimore

APPENDIX E PROGRAM LISTINGS

TITLE	PAGE
MAIN1	210
MAIN2	218
MAIN3	223
DEFINE	228
READ1	230
SORT3	235
LCOST2	238
TRANS	245
NETOUT	255
SINGL3	258

MAIN1, MAIN2, and MAIN3 contain the instructions for the input routine and the calculations for the input values to the network algorithms. The reader is referred to Chapter 3 for a discussion of the network algorithms and their required inputs. The MAIN1 routine chains to the READ1 routine, and READ1 serves as the main program for the network algorithms. The flow-dependent algorithm is contained in LCOST2, and the transshipment algorithm is contained in TRANS. NETOUT provides output for both algorithms. SINGL3 examines the path from any one origin to any one destination, and is available to the user with either of the algorithms.


```

(*$$+,I-,N+*)
PROGRAM MAIN1;

USES APPLESTUFF,CHAINSTUFF,(*$U SCREENPORT.CODE*)SCREENPORT,
  (*$U SCREEN.CODE*)SCREEN,(*$U JOHNSTUFF.CODE*)
  JOHNSTUFF;
  (*1 printer:*)
VAR
  a,i,j,f,
  h,
  iterate, moves,temp      : integer;
  y : interactive;
  x,z : char;
  tshipr,minepr : packed array[1..4] of real;
  trukpr : packed array[1..2] of real;
  fprtpr : real;
  tshipr,railpr : packed array[1..5] of real;
  frailpr,portpr,veslpr : packed array[1..3] of real;
  converge : real;
  numlink : string;
  netonode,netdnode,netflag,
  netc0,netc1,netc2,
  netcap,netsup : packed array[1..250] of integer;
  minefil : packed array[1..16,1..10] of integer;
  railfil : packed array[1..16,1..12] of integer;

  (* counter for link data *)
  (* parameter variables *)
  (* link data variables *)

segment PROCEDURE PARAMOUT;
  (* loads parameters and writes them
  to file for use by main2 & main3 *)

var tsfile, ptfile,vslfile,fptfile,ftsfile,frlfile : file of real;

BEGIN
  mark(heap);
  loadform('coall:param.form',hf,txtp,p,fcount,icount,ioerr);
  for i := 1 to 4 do begin
    minepr[i] := p^.rval; p := p^.n; end;
  for i := 1 to 2 do begin
    trukpr[i] := p^.rval; p := p^.n; end;
  for i := 1 to 5 do begin
    railpr[i] := p^.rval; p := p^.n; end;
  rewrite(tsfile,'coall:tsfile.data');
  for i := 1 to 5 do begin
    tshipr[i] := p^.rval; tsfile^ := tshipr[i]; put(tsfile);
    p := p^.n; end; close(tsfile,lock);
  rewrite(ptfile,'coall:ptfile.data');
  for i := 1 to 3 do begin
    portpr[i] := p^.rval; ptfile^ := portpr[i]; put(ptfile);
    p := p^.n; end; close(ptfile,lock);

```

```

rewrite(vslfile,'coall:vslfile.data');
  for i := 1 to 3 do begin
    veslpr[i] := p^.rval; vslfile^ := veslpr[i]; put(vslfile);
    p := p^.n; end; close(vslfile,lock);
rewrite(fptfile,'coall:fptfile.data');
  begin fprtpr := p^.rval; fptfile^ := fprtpr; put(fptfile);
    p := p^.n; close(fptfile,lock); end;
rewrite(ftsfile,'coall:ftsfile.data');
  for i := 1 to 4 do begin
    ftshipr[i] := p^.rval; ftsfile^ := ftshipr[i]; put(ftsfile);
    p := p^.n; end; close(ftsfile,lock);
rewrite(frlfile,'coall:frlfile.data');
  for i := 1 to 3 do begin
    frailpr[i] := p^.rval; frlfile^ := frailpr[i]; put(frlfile);
    p := p^.n; end; close(frlfile,lock);
iterate := p^.ival; p := p^.n;
moves := p^.ival; p := p^.n;
converge := p^.rval;
END; (* paramout *)

```

segment PROCEDURE PARAMS;

(* loads parameter screen *)

```

BEGIN
  ff := 'coall:param.form';
  showit;
  saveit;
  paramout;
  draw;
END;

```

segment PROCEDURE MINE1;

(* calculates costs & price *)

```

BEGIN
  i := 1;
  while (minefil[i,1] > 0) and (i < 16) do begin
    netonode[h] := minefil[i,1];
    netdnode[h] := minefil[i,2];
    netflag[h] := 1;
    if (minefil[i,7] <= 15) then begin
      if (minefil[i,10] = 1) then begin
        netc0[h] := 3000;
        netc1[h] := ROUND(100 * 0.086 * minefil[i,6]);
        if (minefil[i,3] = 0) then minefil[i,3] := netc0[h] +
          netc1[h] * minefil[i,6]; end
      else begin
        netc0[h] := 3200;
      end
    end
  end

```

(* low sulfur *)

(* region 1 *)

(* region 3 *)

```

    netc1[h] := ROUND(100 * 0.5 * minefil[i,6]);
    if (minefil[i,3] = 0) then minefil[i,3] := netc0[h] +
        netc1[h] * minefil[i,6]; end; end
else begin
    if (minefil[i,10] = 2) then begin
        netc0[h] := 2900;
        netc1[h] := ROUND(100 * 0.0275 * minefil[i,6]);
        if (minefil[i,3] = 0) then minefil[i,3] := netc0[h] +
            netc1[h] * minefil[i,6]; end; end
    else begin
        netc0[h] := 2300;
        netc1[h] := ROUND(100 * 0.35 * minefil[i,6]);
        if (minefil[i,3] = 0) then minefil[i,3] := netc0[h] +
            netc1[h] * minefil[i,6]; end; end;
    netc2[h] := ROUND(100 * minepr[4]);
    netcap[h] := 16383;
    netsup[h] := minefil[i,5];
    if (minefil[i,4] < 0) then minefil[i,4] := ROUND(minepr[1] +
        minepr[2] * minefil[i,3] + minepr[3] * minefil[i,7]);
    h := h + 1; i := i + 1; end;
draw;
END;

segment PROCEDURE MINES;
BEGIN
    ff := 'coall:usmine.form';
    showit;
    for i := 1 to 15 do begin
        p := p^.n;
        for j := 1 to 10 do begin
            if p^.ival < 0 then minefil[i,j] := 0 else
                minefil[i,j] := p^.ival; p := p^.n; end; end;
    saveit;
    minel;
END;

segment PROCEDURE UTILITY;
(* writes values to an array
and computes costs *)
var utilfil : packed array[1..16,1..10] of integer;
BEGIN
    ff := 'coal2:utility.form';
    showit;
    for i := 1 to 15 do begin
        p := p^.n;

```

```

    for j := 1 to 10 do begin
        if p^.ival < 0 then utilfil[i,j] := 0 else
            utilfil[i,j] := p^.ival; p := p^.n; end; end;
saveit;
i := 1;
while (utilfil[i,1] > 0) and (i < 16) do begin
    netonode[h] := utilfil[i,1];
    netdnode[h] := utilfil[i,2];
    netflag[h] := 3;
    if (utilfil[i,3] = 0) then
        if utilfil[i,7] < 0 then utilfil[i,3] := 2 else
            utilfil[i,3] := utilfil[i,7];
    netc0[h] := utilfil[i,3] * 100;
    netc1[h] := 0; netc2[h] := 1;
    netcap[h] := 16383;
    netsup[h] := utilfil[i,6];
    if (utilfil[i,4] < 0) then utilfil[i,4] := 0;
    h := h + 1; i := i + 1; end;
draw;
END;    (* utility *)

segment PROCEDURE TRUCK;                                (* writes to an array
                                                         and computes costs *)

var trukfil : packed array[1..16,1..6] of integer;

BEGIN
    ff := 'coal2:truck.form';
    showit;
    for i := 1 to 15 do begin
        p := p^.n;
        for j := 1 to 6 do begin
            if p^.ival < 0 then trukfil[i,j] := 0 else
                trukfil[i,j] := p^.ival; p := p^.n; end; end;
saveit;
i := 1;
while (trukfil[i,1] > 0) and (i < 16) do begin
    netonode[h] := trukfil[i,1];
    netdnode[h] := trukfil[i,2];
    netflag[h] := 2;
    if (trukfil[i,3] > 0) then netc0[h] := trukfil[i,3] * 100
        else netc0[h] := ROUND((trukpr[1] + trukpr[2] * trukfil[i,5] / 20)*100);
    netc1[h] := 0; netc2[h] := 1;
    netcap[h] := 16383;
    netsup[h] := 0;
    if (trukfil[i,4] < 0) then trukfil[i,4] := trukfil[i,3];
    h := h + 1; i := i + 1; end;
draw;
END;

```

segment PROCEDURE RAIL2;

BEGIN

netc2[h] := ROUND(railpr[5] * 100);
netcap[h] := 16383;
netsup[h] := 0;

if (railfil[i,4] < 0) then railfil[i,4] := ROUND(railpr[1] * railpr[2]
* railfil[i,3] + railpr[3] * railfil[i,10] +
railpr[4] * railfil[i,5]);

END;

segment PROCEDURE RAIL1;

(* computes rail costs for
double track shipments *)

BEGIN

(* double track *)

if railfil[i,9] = 0 then begin (* single car shipment *)

netc0[h] := (3 * railfil[i,5]);

if (railfil[i,12] = 3) and (railfil[i,11] > 38)

then netc1[h] := ROUND(1.6 * railfil[i,5]) else

netc1[h] := 0;

(* manual signal *)

if (railfil[i,12] = 4) and (railfil[i,11] > 64)

then netc1[h] := ROUND(0.94 * railfil[i,5]) else

netc1[h] := 0;

(* ABS *)

if (railfil[i,12] = 5) and (railfil[i,11] > 102)

then netc1[h] := ROUND(0.53 * railfil[i,5]) else

netc1[h] := 0; (* CTC *) end (* single car *)

else begin

(* unit train shipment *)

netc0[h] := ROUND(1.6 * railfil[i,5]);

if (railfil[i,12] = 3) and (railfil[i,11] > 40)

then netc1[h] := ROUND(1.4 * railfil[i,5]) else

netc1[h] := 0;

(* manual signal *)

if (railfil[i,12] = 4) and (railfil[i,11] > 64)

then netc1[h] := ROUND(0.65 * railfil[i,5]) else

netc1[h] := 0;

(* ABS *)

if (railfil[i,12] = 5) and (railfil[i,11] > 110)

then netc1[h] := ROUND(0.53 * railfil[i,5]) else

netc1[h] := 0; (* CTC *) end; (* unit train shipment *)

if railfil[i,3] > 0 then begin

netc0[h] := (railfil[i,3] * 100);

netc1[h] := 0; end;

rail2;

END; (* rail *)

segment PROCEDURE RAIL;

(* write values to an array and calculates costs for single track shipments *)

BEGIN

showit;

for i := 1 to 15 do begin

 p := pⁿ;

 for j := 1 to 12 do begin

 if p^{.ival} < 0 then railfil[i,j] := 0 else

 railfil[i,j] := p^{.ival}; p := pⁿ; end; end;

saveit;

i := 1;

while (railfil[i,1] > 0) and (i < 16) do begin

 netonode[h] := railfil[i,1];

 netdnode[h] := railfil[i,2];

 netflag[h] := 2;

 if (railfil[i,3] = 0) then begin

 if (railfil[i,12] < 3) then begin

 (* single track *)

 if railfil[i,9] = 0 then begin

 (* single car shipment *)

 netc0[h] := (3 * railfil[i,5]);

 if (railfil[i,12] = 0) and (railfil[i,11] > 12)

 then netc1[h] := ROUND(4.6 * railfil[i,5]) else

 netc1[h] := 0;

 (* manual signal *)

 if (railfil[i,12] = 1) and (railfil[i,11] > 18)

 then netc1[h] := ROUND(2.9 * railfil[i,5]) else

 netc1[h] := 0;

 (* ABS *)

 if (railfil[i,12] = 2) and (railfil[i,11] > 39)

 then netc1[h] := ROUND(1.5 * railfil[i,5]) else

 netc1[h] := 0;

 (* CTC *)

 end (* single car *)

 else begin

 (* unit train shipment *)

 netc0[h] := ROUND(1.6 * railfil[i,5]);

 if (railfil[i,12] = 0) and (railfil[i,11] > 12)

 then netc1[h] := ROUND(5.6 * railfil[i,5]) else

 netc1[h] := 0;

 (* manual signal *)

 if (railfil[i,12] = 1) and (railfil[i,11] > 19)

 then netc1[h] := ROUND(2.4 * railfil[i,5]) else

 netc1[h] := 0;

 (* ABS *)

 if (railfil[i,12] = 2) and (railfil[i,11] > 41)

 then netc1[h] := ROUND(1.3 * railfil[i,5]) else

 netc1[h] := 0; end;

 (* CTC *)

 end (* unit train *)

 end; RAIL1; h := h + 1; i := i + 1; end;

END;

 (* usrail *)

segment PROCEDURE GETRAIL;

(* selects the rail screens *)

BEGIN

ff := 'coal2:rail1.form'; rail; bottom; read(x);

if (x = 'y') or (x = 'Y') then begin

```

        ff := 'coal2:rail2.form'; rail; bottom; read(x);
        if (x = 'y') or (x = 'Y') then begin
            ff := 'coal2:rail3.form'; rail; end; end;
    draw;
END;

segment PROCEDURE FINDIT;                                (* selects the proper screens *)

BEGIN
    draw;
    wipe; write('Get model parameter file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; params; end
    else begin setit; paramout; wipe; end;
    write('Get U. S. mine input file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; mines; end; wipe;
    write('Get utility input file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; utility; end; wipe;
    write('Get U. S. truck input file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; truck; end; wipe;
    write('Get U. S. rail input file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; getrail; end;
END;

segment PROCEDURE OUTPT1;                                (* writes values to a data file *)

var onefile,prfile : file of integer;

BEGIN
    rewrite(onefile, 'coal1:onefile.data');
    for j := 1 to h do begin
        onefile^ := netonode[j]; put(onefile);
        onefile^ := netdnode[j]; put(onefile);
        onefile^ := netflag[j]; put(onefile);
        onefile^ := netc0[j]; put(onefile);
        onefile^ := netc1[j]; put(onefile);
        onefile^ := netc2[j]; put(onefile);
        onefile^ := netcap[j]; put(onefile);
        onefile^ := netsup[j]; put(onefile); end;

    close(onefile,lock);
    rewrite(prfile,'coal1:prfile.data');
    prfile^ := iterate; put(prfile);
    prfile^ := moves; put(prfile);
    prfile^ := ROUND(converge * 100); put(prfile);
    close(prfile,lock);
END;

```

```
BEGIN                                (* main1 *)
  hf := 'dummy';
  h := 1; swapon; FINDIT;
  wipe; write('Computing link information . . . please wait ');
  outpt1;
  swapoff;
  istr(h,2,numlink);
  setcval(numlink);
  setchain('coall:main2');
END.
```



```

(*$S++,I-,N+*)
PROGRAM MAIN2;

USES APPLESTUFF,CHAINSTUFF,(*$U SCREENPORT.CODE*)SCREENPORT,
    (*$U SCREEN.CODE*)SCREEN,(*$U JOHNSTUFF.CODE*)
    JOHNSTUFF;
(*! printer:*)
VAR
    a,i,j,h,k,temp : integer;
    y : interactive;
    x,z : char;
    netonode,netdnode,netflag,
    netc0,netc1,netc2,
    netcap,netsup : packed array[1..250] of INTEGER;
    tshipr : packed array[1..5] of real;
    portpr,veslpr : packed array[1..5] of real;
    fptpr : real;
    big,form : boolean;
    numlink : string;
    twofile : file of integer;

segment PROCEDURE PARAMIN;                                (* reads in parameters *)

var  tsfile,ptfile,vslfile,fptfile : file of real;

BEGIN
    reset(tsfile,'coall:tsfile.data');
    for i := 1 to 5 do begin
        tshipr[i] := tsfile^; get(tsfile); end; close(tsfile,purge);
    reset(ptfile,'coall:ptfile.data');
    for i := 1 to 3 do begin
        portpr[i] := ptfile^; get(ptfile); end; close(ptfile,purge);
    reset(vslfile,'coall:vslfile.data');
    for i := 1 to 3 do begin
        veslpr[i] := vslfile^; get(vslfile); end; close(vslfile,purge);
    reset(fptfile,'coall:fptfile.data');
    fptpr := fptfile^; close(tsfile,purge);
END;

segment PROCEDURE TSHIP;                                (* loads transshipment screen,
                                                         writes to an array,
                                                         calculates costs *)

var tshipfil : packed array[1..16,1..10] of integer;

BEGIN
    ff := 'coal2:tship.form';
    showit;

```

```

for i := 1 to 15 do begin
  p := p^.n;
  for j := 1 to 10 do begin
    if p^.ival < 0 then tshipfil[i,j] := 0 else
      tshipfil[i,j] := p^.ival; p := p^.n; end; end;
  saveit;
  i := 1;
  while (tshipfil[i,1] > 0) and (i < 16) do begin
    netonode[h] := tshipfil[i,1];
    netdnode[h] := tshipfil[i,2];
    netflag[h] := 2;
    if (tshipfil[i,3] > 0) then netc0[h] := tshipfil[i,3] * 100
      else netc0[h] := round((tshipr[1] + (tshipr[2] * tshipfil[i,7]) +
        (tshipr[3] * tshipfil[i,8]) + (tshipr[4] *
          tshipfil[i,8])) * 100);
    netc1[h] := 0; netc2[h] := 1;
    netcap[h] := round(tshipfil[i,5] * tshipr[5]);
    netsup[h] := 0;
    if (tshipfil[i,4] < 0) then tshipfil[i,4] := tshipfil[i,3];
    h := h + 1; i := i + 1; end;
  draw;
END;      (* tship *)

segment PROCEDURE PORT;      (* loads port screen, writes to
                               an array, calculates costs *)

var portfil : packed array[1..16,1..10] of integer;

BEGIN
  ff := 'coal2:port.form';
  showit;
  for i := 1 to 15 do begin
    p := p^.n;
    for j := 1 to 10 do begin
      if p^.ival < 0 then portfil[i,j] := 0 else
        portfil[i,j] := p^.ival; p := p^.n; end; end;
    saveit;
    i := 1;
    while (portfil[i,1] > 0) and (i < 16) do begin
      netonode[h] := portfil[i,1];
      netdnode[h] := portfil[i,2];
      netflag[h] := 2;
      netc0[h] := 0;
      if (portfil[i,3] = 0) then portfil[i,3] := ((.9 * portfil[i,8] +
        (portfil[i,9] DIV portfil[i,10])) * 0.12);
      netc1[h] := round(100 * (portpr[1] + portpr[2] * portfil[i,3]));
      netc2[h] := round(100 * portpr[3]);
      netcap[h] := 16383;
      netsup[h] := 0;
    end;
  end;

```

```

        if (portfil[i,4] < 0) then portfil[i,4] := round(portpr[1] + portpr[2]
            * portfil[i,3]);
        h := h + 1; i := i + 1; end;
    draw;
END;      (* port *)

segment PROCEDURE VESSEL;                                (* loads vessel screen, writes to
                                                         an array, calculates costs *)

var vesslfil : packed array[1..16,1..10] of integer;

BEGIN
    showit;
    for i := 1 to 15 do begin
        p := p^.n;
        for j := 1 to 10 do begin
            if p^.ival < 0 then vesslfil[i,j] := 0 else
                vesslfil[i,j] := p^.ival; p := p^.n; end; end;
        saveit;
        i := 1;
        while (vesslfil[i,1] > 0) and (i < 16) do begin
            netonode[h] := vesslfil[i,1];
            netdnode[h] := vesslfil[i,2];
            netflag[h] := 2;
            if (vesslfil[i,3] = 0) then vesslfil[i,3] := round(veslpr[1] + veslpr[2]
                * vesslfil[i,8] + (veslpr[3] * vesslfil[i,7] / 100));
            netc0[h] := round(vesslfil[i,3] * 100);
            netc1[h] := 0; netc2[h] := 1;
            netcap[h] := 16383;
            netsup[h] := 0;
            if (vesslfil[i,4] < 0) then vesslfil[i,4] := round(veslpr[1] / 100 *
                vesslfil[i,3]);
            h := h + 1; i := i + 1; end;
        END;      (* vesll *)

segment PROCEDURE GETVSL;                                (* selects vessel screens *)

BEGIN
    ff := 'coal2:vesl1.form'; vessel; bottom; read(x);
    if (x = 'y') or (x = 'Y') then begin
        ff := 'coal2:vesl2.form'; vessel; bottom; read(x);
        if (x = 'y') or (x = 'Y') then begin
            gotoxy(0,23);
            write('Insert COAL3: diskette in place of COAL2: and press return. ');
            beep; readln;
            ff := 'coal3:vesl3.form'; vessel; bottom; read(x);
            if (x = 'y') or (x = 'Y') then begin
                ff := 'coal3:vesl4.form'; vessel; bottom; read(x);
                if (x = 'y') or (x = 'Y') then begin

```

```

        ff := 'coal3:vesl5.form'; vessel; bottom; read(x);
        if (x = 'y') or (x = 'Y') then begin
            ff := 'coal3:vesl6.form'; vessel; end; end; end; end; end;
    draw;
END;

segment PROCEDURE FPORT;                                (* loads foreign port screen,
                                                         writes to an array, calculates costs *)

var fportfil : packed array[1..16,1..10] of integer;

BEGIN
    ff := 'coal3:fport.form';
    showit;
    for i := 1 to 15 do begin
        p := p^.n;
        for j := 1 to 10 do begin
            if p^.ival < 0 then fportfil[i,j] := 0 else
                fportfil[i,j] := p^.ival; p := p^.n; end; end;
    saveit;
    i := 1;
    while (fportfil[i,1] > 0) and (i < 16) do begin
        netonode[h] := fportfil[i,1];
        netdnode[h] := fportfil[i,2];
        netflag[h] := 2;
        if (fportfil[i,3] = 0) then begin
            if fportfil[i,8] = 0 then fportfil[i,3] := 2 else
                fportfil[i,3] := (fportfil[i,8] + fportfil[i,9] DIV
                    fportfil[i,10]); end;
            netc0[h] := round(fportfil[i,3] * 100);
            netc1[h] := 0;
            netc2[h] := 1;
            netcap[h] := 16383;
            netsup[h] := 0;
            if (fportfil[i,4] < 0) then fportfil[i,4] := fportfil[i,3];
            h := h + 1; i := i + 1; end;
    draw;
END;                                (* fport *)

segment PROCEDURE FINDIT;                                (* selects proper screen *)

BEGIN
    write('Get U. S. transshipment input file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; tship; end; wipe;
    write('Get U. S. port input file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; port; end; wipe;
    write('Get vessel input file? . . . '); beep; read(z);
    if (z = 'Y') or (z = 'y') then begin wait; getvsl; end; wipe;

```

```

write('Get foreign port input file? . . . '); beep; read(z);
if (z = 'Y') or (z = 'y') then begin wait; fport; end; wipe;
END;

```

```

segment PROCEDURE OUTPT2;

```

```

(* writes data to files *)

```

```

var onefile : file of integer;
    dumb : integer;

```

```

BEGIN

```

```

    rewrite(twofile,'coall:twofile.data');

```

```

    reset(onefile,'coall:onefile.data');

```

```

    for i := 1 to ((k - 1) * 8) do begin

```

```

        dumb := onefile^;

```

```

        twofile^ := dumb; put(twofile);

```

```

        get(onefile); end;

```

```

    close(onefile,purge);

```

```

    for j := 1 to (h - 1) do begin

```

```

        twofile^ := netonode[j]; put(twofile);

```

```

        twofile^ := netdnode[j]; put(twofile);

```

```

        twofile^ := netflag[j]; put(twofile);

```

```

        twofile^ := netc0[j]; put(twofile);

```

```

        twofile^ := netc1[j]; put(twofile);

```

```

        twofile^ := netc2[j]; put(twofile);

```

```

        twofile^ := netcap[j]; put(twofile);

```

```

        twofile^ := netsup[j]; put(twofile); end;

```

```

    close(twofile,lock);

```

```

END;

```

```

BEGIN

```

```

(* main2 *)

```

```

    swapon;

```

```

    h := 1;

```

```

    hf := 'dummy';

```

```

    paramin;

```

```

    draw;

```

```

    FINDIT;

```

```

    getcval(numlink);

```

```

    int(numlink,big,form,k);

```

```

    temp := (h + k) - 2;

```

```

    outpt2;

```

```

    istr(temp,3,numlink);

```

```

    setcval(numlink);

```

```

    swapoff;

```

```

    setchain('coall:main3');

```

```

END.

```

```

(*$S++,I-,N+*)
PROGRAM MAIN3;

USES APPLESTUFF,CHAINSTUFF,(*$U SCREENPORT.CODE*)SCREENPORT,
    (*$U SCREEN.CODE*)SCREEN,(*$U JOHNSTUFF.CODE*)
    JOHNSTUFF;
(*! printer:*)
VAR
    a,f,g,h,i,j,k : integer;
    x,z : char;
    y : interactive;
    ftshipr : packed array[1..4] of real;          (* parameters from main1 *)
    frailpr : packed array[1..3] of real;
    netonode,netdnode,netflag,
    netc0,netc1,netc2,netcap,netsup : packed array[1..250] of integer;
    gfile,netfile : file of integer;
    numlink : string;
    big,form : boolean;

segment PROCEDURE PARAMIN;                        (* reads in parameters *)

var fptfile,ftsfile,frlfile : file of real;

BEGIN
    reset(ftsfile,'coal1:ftsfile.data');
    for i := 1 to 4 do begin
        ftshipr[i] := ftsfile^; get(ftsfile); end; close(ftsfile,purge);
    reset(frlfile,'coal1:frlfile.data');
    for i := 1 to 3 do begin
        frailpr[i] := frlfile^; get(frlfile); end; close(frlfile,purge);
    END;

segment PROCEDURE FTSHIP;                        (* writes values to an array,
                                                calculates costs *)

var ftshipfil : packed array[1..16,1..10] of integer;

BEGIN
    ff := 'coal3:ftship.form';
    showit;
    for i := 1 to 15 do begin
        p := p^.n;
        for j := 1 to 10 do begin
            if p^.ival < 0 then ftshipfil[i,j] := 0 else
                ftshipfil[i,j] := p^.ival; p := p^.n; end; end;
        saveit;
        i := 1;
    end;

```

```

while (ftshipfil[i,1] > 0) and (i < 16) do begin
  netonode[h] := ftshipfil[i,1];
  netdnode[h] := ftshipfil[i,2];
  netflag[h] := 2;
  if (ftshipfil[i,3] > 0) then netc0[h] := ftshipfil[i,3] * 100
  else netc0[h] := round((ftshipr[1] + (ftshipr[2] * ftshipfil[i,7]) +
    (ftshipr[3] * ftshipfil[i,8]) + (ftshipr[4] *
    ftshipfil[i,8])) * 100);
  netc1[h] := 0; netc2[h] := 1;
  netcap[h] := ftshipfil[i,5];
  netsup[h] := 0;
  if (ftshipfil[i,4] < 0) then ftshipfil[i,4] := ftshipfil[i,3];
  h := h + 1; i := i + 1; end;
draw;
END;          (* ftship *)

segment PROCEDURE FRAIL;          (* writes values to an array,
                                   calculates costs *)

var frailfil : packed array[1..16,1..12] of integer;

BEGIN
  ff := 'coal3:frail.form';
  showit;
  for i := 1 to 15 do begin
    p := p^.n;
    for j := 1 to 12 do begin
      if p^.ival < 0 then frailfil[i,j] := 0 else
        frailfil[i,j] := p^.ival; p := p^.n; end; end;
  saveit;
  i := 1;
  while (frailfil[i,1] > 0) and (i < 16) do begin
    netonode[h] := frailfil[i,1];
    netdnode[h] := frailfil[i,2];
    netflag[h] := 2;
    if (frailfil[i,3] = 0) then frailfil[i,3] := round((frailpr[1] +
      frailpr[2] * frailfil[i,5]) * 100);
    netc0[h] := frailfil[i,3];
    netc1[h] := 0;
    netc2[h] := 1;
    netcap[h] := 16383;
    netsup[h] := 0;
    if (frailfil[i,4] < 0) then frailfil[i,4] := round(frailfil[i,3]
      * frailpr[3]);
    h := h + 1; i := i + 1; end;
  draw;
END;          (* frail *)

```

segment PROCEDURE ODFILE;

(* loads o-d flow screen,
stores values *)

var flwfile : file of integer;
orig,dest : packed array[1..16] of integer;
odfil : packed array[1..16,1..16] of integer;

BEGIN

ff := 'coal3:odflow.form';
showit;
for i := 1 to 15 do begin
if p^.ival < 0 then dest[i] := 0 else
dest[i] := p^.ival; p := p^.n; end;
for i := 1 to 15 do begin
if p^.ival < 0 then orig[i] := 0 else
orig[i] := p^.ival; p := p^.n; end;

g := 0; i := 1; j := 1;
while (orig[i] > 0) and (i < 16) do begin
while (dest[j] > 0) and (j < 16) do begin
odfil[i,j] := p^.ival; p := p^.n;
g := g + 1; j := j + 1; end; j := 1;
i := i + 1; end;

saveit;

rewrite(gfile, 'coal1:gfile.data'); (* gfile keeps the total # of flows *)
gfile^ := g; put(gfile); (* h is the number of links *)
gfile^ := ((h + k) - 1); put(gfile);
close(gfile,lock);

i := 1; j := 1;
rewrite(flwfile, 'coal1:flwfile.data'); (* flwfile contains the flows *)
while (orig[i] > 0) and (i < 16) do begin
while (dest[j] > 0) and (j < 16) do begin
flwfile^ := orig[i]; put(flwfile);
flwfile^ := dest[j]; put(flwfile);
flwfile^ := odfil[i,j]; put(flwfile); j := j + 1; end; j := 1;
i := i + 1; end;
close(flwfile,lock);
draw;
END;

segment PROCEDURE OUTPT2;

(* writes network information *)

var twofile : file of integer;
dumb : integer;

BEGIN

```
rewrite(netfile,'coall:netfile.data');
reset(twofile,'coall:twofile.data');
for i := 1 to (k * 8) do begin
    dumb := twofile^;
    netfile^ := dumb; put(netfile);
    get(twofile); end;
close(twofile,purge);

for j := 1 to (h - 1) do begin
    netfile^ := netonode[j]; put(netfile);
    netfile^ := netdnode[j]; put(netfile);
    netfile^ := netflag[j]; put(netfile);
    netfile^ := netc0[j]; put(netfile);
    netfile^ := netc1[j]; put(netfile);
    netfile^ := netc2[j]; put(netfile);
    netfile^ := netcap[j]; put(netfile);
    netfile^ := netsup[j]; put(netfile); end;
close(netfile,lock);
END;
```

procedure test;forward;

segment PROCEDURE FINDIT;

(* selects the proper screens *)

BEGIN

```
write('Get foreign transshipment input file? . . . '); beep; read(z);
if (z = 'Y') or (z = 'y') then begin wait; ftship; end; wipe;
write('Get foreign rail input file? . . . '); beep; read(z);
if (z = 'Y') or (z = 'y') then begin wait; frail; end; wipe;
write('Get O - D flow file? . . . '); beep; read(z);
if (z = 'Y') or (z = 'y') then begin wait; odfile; end; wipe;
write('Run network algorithm? . . . '); beep; read(z);
if (z = 'y') or (z = 'Y') then begin wipe; outpt2; page(output);
    write('Computing network flows . . . output will follow');
    setchain('coall:netwk'); end
else begin wipe; write('Exit program? . . . '); beep; read(z);
    if (z = 'y') or (z = 'Y') then begin outpt2; exit(findit); end
    else setchain('coall:main1'); end;
END; (* findit *)
```

procedure test;

var yoyo : integer;

```

begin
  reset(netfile,'coall:netfile.data');
  for i := 1 to ((h + k)-1) do begin
    for j := 1 to 8 do begin
      yoyo := netfile^; write(y,yoyo,' ');
      get(netfile); end; writeln(y); end;
    close(netfile,lock);

    reset(gfile,'coall:gfile.data');
    yoyo := gfile^; writeln(y,' g = ',yoyo); get(gfile);
    yoyo := gfile^; writeln(y,' h = ',yoyo);
    close(gfile,lock);
  end;

```

```

BEGIN          (* main3 *);
  h := 1;
  swapon;
  rewrite(y,'printer:');
  hf := 'dummy';
  paramin;
  getcval(numlink);
  int(numlink,big,form,k);
  draw;
  FINDIT;
  test;
  swapoff;
  close(y);
END.

```

DEFINITIONS

INPUTS - THIS LIST INCLUDES ALL VALUES PASSED FROM THE INPUT SCREENS TO THE MAIN COAL NETWORK PROGRAM. IT ALSO INCLUDES THOSE VARIABLES WHICH ARE CHANGED TO THE PROPER TYPE DURING THE INPUT PROCESS.

1. ONODE - HEAD NODE OF ANY LINK IN THE NETWORK, INTEGER VALUE
2. DNODE - TAIL NODE OF ANY LINK IN THE NETWORK, INTEGER VALUE
3. FLAG - SIGNAL TO POSITION A LINK IN THE NETWORK, FLAG = 1 MEANS THE LINK CONTAINS AN ORIGIN, FLAG = 3 MEANS THE LINK CONTAINS A DESTINATION, FLAG = 2 STANDS FOR AN INTERMEDIATE LINK, INTEGER VALUE
4. C0 - FIXED PORTION OF LINK COST, INTEGER VALUE
5. C10 - MULTIPLIER IN LINK COST EQUATION AS RECEIVED FROM INPUT SCREEN, INTEGER VALUE FROM INPUT SCREEN
6. C1 - MULTIPLIER IN LINK COST EQUATION USED IN FLOW-DEPENDENT ALGORITHM, COMPUTED FROM INPUT VALUE C10, INTEGER VALUE
7. C20 - EXPONENT IN LINK COST EQUATION AS RECEIVED FROM INPUT SCREEN, INTEGER VALUE FROM INPUT SCREEN
8. C2 - EXPONENT IN LINK COST EQUATION USED IN FLOW-DEPENDENT ALGORITHM, COMPUTED FROM INPUT VALUE C20, REAL VALUE
9. CP - LINK CAPACITY, INTEGER VALUE
10. SUP - NODE SUPPLY OR DEMAND FROM INPUT SCREEN, ORIGINS HAVE SUPPLY, DESTINATIONS HAVE DEMAND, INTERMEDIATE NODES HAVE ZERO SUPPLY, NO REQUIREMENT FOR SUPPLY TO EQUAL DEMAND, INTEGER VALUE
11. ORIGIN - NETWORK ORIGIN, REPRESENTS A MINE, INTEGER VALUE
12. DESTIN - NETWORK DESTINATION, REPRESENTS A UTILITY, INTEGER VALUE
13. FLOWIN - ORIGIN TO DESTINATION FLOW BETWEEN A SPECIFIC O-D PAIR, USED IN FLOW-DEPENDENT ALGORITHM, NO RELATION TO SUP, INTEGER VALUE
14. ITER - CONTROL PARAMETER TO LIMIT THE NUMBER OF ITERATIONS OF LINK COST COMPUTATIONS, DEFAULT IS 2, INTEGER VALUE
15. MSNO - CONTROL PARAMETER TO LIMIT THE NUMBER OF ITERATIONS OF WEIGHTING FACTOR FOR FLOWS, DEFAULT IS 1, INTEGER VALUE
16. RES - CONTROL PARAMETER PASSED BY THE INPUT SCREEN TO THE CONVERGENCE CRITERION FOR THE FLOW-DEPENDENT ALGORITHM, INTEGER VALUE, DEFAULT IS 2
17. RESULT - CONTROL PARAMETER TO DETERMINE CONVERGENCE REQUIREMENTS, CHANGED DURING INPUT TO A REAL VALUE, DEFAULT IS 0.02
18. NLINK - NUMBER OF LINKS ENTERED BY THE USER, USED TO CONTROL THE INPUT OF THE NETWORK STRUCTURE, INTEGER VALUE
19. G - NUMBER OF ORIGIN-DESTINATION PAIRS ENTERED BY USER, USED TO CONTROL INPUT OF O-D FLOW MATRIX, INTEGER VALUE

VARIABLES - THESE ARE THE VARIABLES GENERATED BY THE ALGORITHMS IN
THE SOLUTION OF THE NETWORK

1. FLOW - MEASURE OF LINK COAL VOLUME USED IN THE FLOW-DEPENDENT ALGORITHM, A BASIC OUTPUT OF THE NETWORK
2. ICOST - UNIT COST OF SHIPPING A TON OF COAL ON A LINK IN THE NETWORK, A BASIC OUTPUT OF THE FLOW-DEPENDENT ALGORITHM, USED AS THE FIXED LINK COSTS FOR THE TRANSSHIPMENT ALGORITHM
3. TCFLOW - MEASURE OF THE TOTAL COST OF THE FINAL NETWORK FLOW PATTERN IN THE FLOW-DEPENDENT ALGORITHM
4. TCTRAN - MEASURE OF THE TOTAL COST OF THE FINAL NETWORK FLOW PATTERN IN THE TRANSSHIPMENT ALGORITHM
5. S - NODE SUPPLY ARRAY USED IN THE TRANSSHIPMENT ALGORITHM
6. X - MEASURE OF LINK FLOWS IN THE TRANSSHIPMENT ALGORITHM WHEN THE LINK IS PROPERLY ORIENTED
7. CPX - MEASURE OF LINK FLOWS IN THE TRANSSHIPMENT ALGORITHM WHEN THE LINK IS ORIENTED BACKWARDS
8. P - PREDECESSOR ARRAY IN THE FLOW-DEPENDENT ALGORITHM
9. PB - INDIVIDUAL O-D PATH FOUND FROM THE PREDECESSOR ARRAY IN THE FLOW-DEPENDENT ALGORITHM
10. PN - PREDECESSOR ARRAY IN THE TRANSSHIPMENT ALGORITHM
11. L - ARRAY WHICH HOLDS THE LOWEST COST FROM A GIVEN ORIGIN TO THE NODE ALONG THE LOWEST-COST PATH IN FLOW-DEPENDENT ALGORITHM
12. SI - LABELING ARRAY TO DETERMINE WHICH NODES TO INVESTIGATE IN DETERMINING THE LOWEST-COST O-D PATHS IN FLOW-DEPENDENT ALGORITHM
13. H - ARRAY WHICH REPLACES THE ONODE ARRAY AND CONTAINS POINTERS TO THE DNODE ARRAY TO INDICATE THE DNODES FROM A GIVEN ONODE, USED IN TRANSSHIPMENT ALGORITHM
14. D - DEPTH OF NODE IN SOLUTION TREE OF THE TRANSSHIPMENT ALGORITHM
15. IT - PREORDER TRAVERSAL THREAD USED IN THE UPDATE OF THE SOLUTION TREE VARIABLES IN THE TRANSSHIPMENT
16. U - NODE POTENTIAL IN THE TRANSSHIPMENT ALGORITHM, USED TO DETERMINE THE LINKS WHICH WILL ENTER AND LEAVE THE SOLUTION BASIS
17. TOTS - MEASURE OF THE TOTAL SUPPLY AVAILABLE FROM ALL SOURCES IN THE TRANSSHIPMENT ALGORITHM
18. TODD - MEASURE OF THE TOTAL DEMAND REQUIRED BY ALL DESTINATION UTILITIES IN THE TRANSSHIPMENT ALGORITHM

```

$USES USINGL3 IN #10:SINGL3.CODE OVERLAY
$USES UNETOUT IN #10:NETOUT.CODE OVERLAY
$USES UTRANS IN #10:TRANS.CODE OVERLAY
$USES ULCOST2 IN #10:LCOST2.CODE OVERLAY
$USES USORT3 IN #10:SORT3.CODE

```

```

PROGRAM READ1
CHARACTER*1 CHOICE,ANS,YES,NO
INTEGER ONODE(300),DNODE(300),FLAG(300),ORIGIN(215),DESTIN(215),
-   ITER,MSNO,MAXN,NARC,H(125),KNT,C20(215),PN(215),
-   CPX(215),S(215),MAX,TOTS,TOTD,CP(300),X(215),
-   ICOST(215),FLOWIN(215),FLOW(215),C0(300),C1(300),KEY,TCFLOW,
-   RES,SUP(215),G,NLINK,NUM
REAL C2(300),RESULT,TOTAL

DATA YES,NO/'Y','N'/
DATA MAX,MAXN,NARC,TOTS,TOTD/16383,0,0,0,0/
OPEN(6,FILE='PRINTER:')
OPEN(32,FILE='COAL1:NETFILE.DATA',FORM='UNFORMATTED')
OPEN(33,FILE='COAL1:PRFILE.DATA',FORM='UNFORMATTED')
OPEN(31,FILE='COAL1:GFILE.DATA',FORM='UNFORMATTED')
OPEN(38,FILE='COAL1:FLWFILE.DATA',FORM='UNFORMATTED')
REWIND(31)
REWIND(32)
REWIND(33)
REWIND(38)

WRITE(*,880)
880 FORMAT(/,2X,' WHICH ALGORITHMS DO YOU WISH TO EXECUTE ?',//,5X,
-   'ENTER',/,10X,'1 FOR FLOW-DEPENDENT ONLY',/,10X,
-   '2 FOR TRANSSHIPMENT ONLY',/,10X,
-   '3 FOR BOTH ALGORITHMS',/,5X,'MAKE YOUR SELECTION ',
-   'AND PRESS RETURN',/)

READ(*,881) NUM
881 FORMAT(BN,I2)

IF(NUM .EQ. 2) THEN
WRITE(*,882)
882 FORMAT(/,5X,'HAVE YOU SET THE FIXED COSTS FOR EVERY LINK ',
-   'DURING INPUT? ENTER Y OR N AND PRESS RETURN',/)

READ(*,883) ANS
883 FORMAT(A1)

IF(ANS .NE. YES) NUM = 3

ENDIF

```

C READ IN PARAMETERS. NLINK = NUMBER OF LINKS IN NETWORK, REQUIRED TO
 C CONTROL NETWORK STRUCTURE INPUT. G = NUMBER OF O-D PAIRS, REQUIRED TO
 C CONTROL O-D FLOW INPUT FOR THE FLOW-DEPENDENT ALGORITHM. ITER, MSNO,
 C AND RES ARE CONVERGENT PARAMETERS IN THE FLOW-DEPENDENT ALGORITHM.

READ(31) G, NLINK
 READ(33) ITER, MSNO, RES

C INPUT THE NETWORK STRUCTURE. ONODE IS THE HEAD NODE, DNODE IS THE TAIL
 C NODE. FLAG DEFINES THE POSITION OF LINK IN THE NETWORK - FLAG = 1 FOR
 C MINES, FLAG = 3 FOR UTILITIES, FLAG = 2 FOR ALL INTERMEDIATE LINKS. CO,
 C C10, C20 ARE COST PARAMETERS FOR LINK COSTS. CP IS THE LINK CAPACITY.
 C SUP IS THE AVAILABLE SUPPLY AT THE MINE OR THE DEMAND AT THE UTILITY.
 C SUP = 0 FOR ALL INTERMEDIATE NODES.

DO 900 J2 = 1, NLINK
 READ(32) ONODE(J2)
 READ(32) DNODE(J2)
 READ(32) FLAG(J2)
 READ(32) C0(J2)
 C0(J2) = INT(C0(J2)/100 + .5)
 READ(32) C1(J2)
 C1(J2) = INT(C1(J2)/100 + .5)
 READ(32) C20(J2)
 C2(J2) = REAL(C20(J2)/100)
 READ(32) CP(J2)
 READ(32) SUP(J2)
 900 CONTINUE

C READ IN THE O-D FLOW MATRIX. ORIGIN = MINE, DESTIN = UTILITY,
 C FLOWIN = O-D FLOW. THERE MUST BE AN ENTRY FROM EVERY ORIGIN TO
 C EVERY DESTINATION. ZERO FLOWS ARE ACCEPTABLE AND ARE THE DEFAULT.

DO 910 J3 = 1, G
 READ(38) ORIGIN(J3)
 READ(38) DESTIN(J3)
 READ(38) FLOWIN(J3)
 910 CONTINUE

CLOSE(31, STATUS='DELETE')
 CLOSE(32, STATUS='DELETE')
 CLOSE(33, STATUS='DELETE')
 CLOSE(38, STATUS='DELETE')

C INITIALIZE ALL VARIABLES AND ARRAYS.

```
KEY = 0
KNT = 0
NNODE = 0
RESULT = REAL(RES/100)
DO 901 KI=1,215
  S(KI) = 0
  H(KI) = 0
  IF(KI .GT. NLINK) THEN
    ONODE(KI) = 0
    FLAG(KI) = 0
    ICOST(KI) = 0
    CO(KI) = 0
    CI(KI) = 0
    C2(KI) = 1.0
    CP(KI) = MAX
  ENDIF
```

C COUNT NUMBER OF NODES IN REAL NETWORK (NNODE)
NNODE = MAX0 (NNODE,ONODE(KI),DNODE(KI))

901 CONTINUE

C ARTIFICIAL NODE IS DESIGNATED AS MAXN.

MAXN = NNODE + 1

```
DO 21 I = 1,NNODE
DO 22 J = 1,NLINK
```

```
IF(FLAG(J) .EQ. 1 .AND. ONODE(J) .EQ. I) THEN
  S(I) = SUP(J)
```

```
ELSEIF(FLAG(J) .EQ. 3 .AND. DNODE(J) .EQ. I) THEN
  S(I) = - SUP(J)
```

ENDIF

22 CONTINUE

C COUNT NUMBER OF ARTIFICIAL LINKS

IF(S(I) .NE. 0) KNT = KNT + 1

21 CONTINUE

DO 23 I2= 1, NNODE
II= NLINK + I2

C FORM AND INITIALIZE ARTIFICIAL LINKS FROM SUPPLY NODES (ORIGINS).

IF(S(I2) .GT. 0) THEN
ONODE(II) = I2
DNODE(II) = MAXN
CO(II) = 0
C1(II) = 0
C2(II) = 1.0
CP(II) = 10000
TOTS = TOTS + S(I2)
FLAG(II) = 10

C FORM AND INITIALIZE ARTIFICIAL LINKS TO DEMAND NODES (DESTINATIONS).

ELSEIF (S(I2) .LT. 0) THEN
ONODE(II) = MAXN
DNODE(II) = I2
CO(II) = MAX -1
C1(II) = 0
C2(II) = 1.0
CP(II) = 10000
TOTD = TOTD - S(I2)
FLAG(II) = 10

ENDIF

23 CONTINUE

C TOTAL NUMBER OF LINKS = TOTAL REAL LINKS + TOTAL ARTIFICIAL LINKS.

NARC = NLINK + KNT

CALL SORT3 (ONODE, DNODE, CO, C1, C2, FLAG, NLINK, NARC, MAXN, H, CP)

IF(NUM .EQ. 1 .OR. NUM .EQ. 3) THEN

- CALL LCOST2(ONODE, DNODE, ICOST, FLAG, NLINK, ORIGIN, DESTIN, FLOW,
FLOWIN, CO, C1, C2, ITER, MSNO, RESULT, NARC, MAXN, KEY, TCFLOW, G)

- CALL NETOUT(ONODE, DNODE, ICOST, FLOW, PN, X, CPX, KEY, NARC, NNODE,
MAXN, TCFLOW, CP)

ENDIF

IF(NUM .EQ. 2 .OR. NUM .EQ. 3) THEN

KEY = 2

CALL TRANS(H, DNODE, X, CPX, PN, S, CP, MAX, MAXN, NARC, ICOST)

- CALL NETOUT(ONODE, DNODE, ICOST, FLOW, PN, X, CPX, KEY, NARC, NNODE,
MAXN, TCFLOW, CP)

ENDIF


```
DO 1071 I=1,NARC
WRITE(*,1070)
WRITE(6,1070)
```

```
C THE SINGL2 SUBROUTINE CAN DETERMINE THE TOTAL COST AND PATH TO
C SHIP COAL FROM ONE ORIGIN TO ONE DESTINATION. THIS SUBROUTINE
C WORKS WITH CONSTANT LINK COSTS, AND, THEREFORE, IT ONLY MAKES
C SENSE TO USE IT AFTER EQUILIBRIUM HAS BEEN REACHED ON THE
C NETWORK AS A WHOLE. THIS SUBROUTINE ASKS FOR THE SPECIFIC ORIGIN
C AND THE SPECIFIC DESTINATION. IT PRINTS OUT THE ORIGIN THE
C DESTINATION AND THE INTERMEDIATE NODES, AS WELL AS THE TOTAL
C COST. THIS COST IS THE UNIT COST FOR A MILLION TONS OF COAL.
```

```
1070 FORMAT(//,2X,'DO YOU WISH TO EXAMINE ANY INDIVIDUAL ROUTES FROM
- A SINGLE ORIGIN TO A SINGLE DESTINATION (Y OR N)?',/)
```

```
      READ(*,1080) CHOICE
1080  FORMAT(A1)
      WRITE(6,1080) CHOICE
      IF(CHOICE .NE. YES) GO TO 7
```

```
      CALL SINGL3(ONODE,DNODE,ICOST,FLAG,NLINK,NARC)
```

```
1071  CONTINUE
```

```
7  STOP
END
```

C THIS SORT ROUTINE USES A HEAPSORT TO REARRANGE THE INPUT ELEMENTS
C IN-PLACE INTO NONDECREASING ORDER

```

SUBROUTINE SORT3 (ONODE,DNODE,CO,C1,C2,FLAG,NLINK,NARC,MAXN,H,CP)
INTEGER F(300),T,FLAG(300),ONODE(300),DNODE(300),H(125),NARC,
- MAXN,CO(300),C1(300),D,E,CP(300)
REAL C2(300),A

```

C COMBINE THE ONODE AND DNODE IN ORDER TO SORT TOGETHER
DO 5000 L1=1,300

C REDESIGNATE UNUSED ONODES TO SEND TO END OF SORT ROUTINE
IF(ONODE(L1) .EQ. 0) ONODE(L1) = MAXN + 1
F(L1) = 200 *(ONODE(L1) -100) + DNODE(L1)
5000 CONTINUE

C FIRST TRANSFORM THE ELEMENTS INTO A HEAP
CALL HEAPIF(F,NARC ,CO,C1,C2,FLAG,CP)

C INTERCHANGE THE NEW MAXIMUM WITH THE ELEMENT AT THE END OF THE TREE
DO 40 I=300 ,2,-1
T=F(I)
F(I)=F(1)
F(1)=T
D=CO(I)
CO(I)=CO(1)
CO(1)=D
E=C1(I)
C1(I)=C1(1)
C1(1)=E
A=C2(I)
C2(I)=C2(1)
C2(1)=A
INT2=FLAG(I)
FLAG(I)=FLAG(1)
FLAG(1)=INT2
INT3 = CP(I)
CP(I) = CP(1)
CP(1) = INT3

C REFORM A SINGLE HEAP WITHOUT THE LAST ELEMENT MOVED IN EACH ITERATION
CALL ADJUST(F,1,I-1,CO,C1,C2,FLAG,CP)

40 CONTINUE
DO 5001 L2=1,300
ONODE(L2) = INT(REAL(F(L2))/200) + 99
DNODE(L2) = MOD(F(L2),200) + 200
5001 CONTINUE

CALL FSTAR(ONODE,NARC,MAXN,H)

RETURN
END

C READJUST THE ELEMENTS OF THE NETWORK INPUT TO FORM A HEAP

```

SUBROUTINE HEAPIF(F,NARC ,C0,C1,C2,FLAG,CP)
REAL C2(300)
INTEGER F(300),FLAG(300),C0(300),C1(300),NARC,CP(300)
DO 140 I=300 /2,1,-1
    CALL ADJUST(F,I,300,C0,C1,C2,FLAG,CP)
140 CONTINUE
RETURN
END

```

C COMBINE TWO BINARY TREES INTO A SINGLE HEAP

```

SUBROUTINE ADJUST(F,I,NARC2 ,C0,C1,C2,FLAG,CP)
REAL C2(215),AA
INTEGER F(215),I,J,N,ITEM,FLAG(215),C0(215),C1(215),DOG,EE,INT2,
- NARC2,INT3,CP(215)
J=I*2
ITEM = F(I)
DOG = C0(I)
EE=C1(I)
AA=C2(I)
INT2 = FLAG(I)
INT3 = CP(I)
C COMPARE LEFT AND RIGHT CHILDREN, POINT J TO THE LARGER CHILD
7 IF(J .LE. NARC2) THEN
    IF(J .LT. NARC2 .AND. F(J) .LT. F(J+1)) J=J+1
    IF(ITEM .GE. F(J)) THEN
C POSITION FOUND FOR THE I-TH CHILD
        GOTO 15
C ELSE MOVE UP THE LARGER CHILD ONE LEVEL IN THE TREE
    ELSE
        F(J/2) = F(J)
        C0(J/2) = C0(J)
        C1(J/2) = C1(J)
        C2(J/2) = C2(J)
        FLAG(J/2) = FLAG(J)
        CP(J/2) = CP(J)
        J=J*2
    ENDIF
    GOTO 7
ENDIF
15 F(J/2) = ITEM
C0(J/2) = DOG
C1(J/2) = EE
C2(J/2) = AA
FLAG(J/2) = INT2
CP(J/2) = INT3
RETURN
END

```

C FORWARD STAR REPLACES LINK-LENGTH ARRAY (ONODE) WITH NODE-LENGTH
C ARRAY (H)

SUBROUTINE FSTAR(ONODE,NARC,MAXN,H)
INTEGER ONODE(300),NARC,MAXN,H(125)

C J COUNTS LINKS; K COUNTS NODES

J=1

K=1

DO 100 KT=1,300

IF(J .LE. NARC .OR. K .LE. MAXN) THEN

IF(K .LT. ONODE(J)) THEN

H(K) = J

K = K + 1

ELSEIF (K .EQ. ONODE(J)) THEN

H(K) = J

J = J + 1

K = K + 1

ELSE

J = J + 1

ENDIF

ENDIF

100 CONTINUE

C ADD A STOPPER TO H TO END CONSIDERATION OF THE ARTIFICIAL NODE
H(MAXN + 1) = NARC + 1

RETURN

END

C FLOW-DEPENDENT CODE DETERMINES THE LOWEST COST PATHS ON A NETWORK
 C WHERE LINK COSTS CHANGE AS THE FLOW VARIES.

```

SUBROUTINE LCOST2(ONODE,DNODE,ICOST,FLAG,NLINK,ORIGIN,DESTIN,FLOW,
-   FLOWIN,CO,C1,C2,ITER,MSNO,RESULT,NARC,MAXN,KEY,TCFLOW,G)
  INTEGER ONODE(300),DNODE(300),FLAG(300),ORIGIN(215),DESTIN(215),
-   NLINK,ITER,MSNO,KEY,NARC,CO(300),C1(300),MAXN,TCFLOW,
-   FLOW(215),FLOWIN(215),ICOST(215),FLOW1(215),FLOWS,DIFF,G
  REAL RESULT,TOTAL,C2(300),ALPHA

```

KEY = 0

C INITIALIZE THE LINK COSTS AS THE FIXED PORTION OF THOSE COSTS

DO 20 M=1,NARC

ICOST(M) = CO(M)

FLOW(M) = 0

20 CONTINUE

C INITIALIZE THE LINK FLOWS BASED ON THE INITIAL LINK COSTS

```

CALL MATRIX(ONODE,DNODE,ICOST,FLAG,NLINK,ORIGIN,DESTIN,FLOW,
-   FLOWIN,RESULT,TOTAL,KEY,NARC,MAXN,TCFLOW,G)

```

DO 200 I=1,ITER

DO 210 J=1,NARC

C SAVE LINK FLOWS FROM PREVIOUS ITERATIONS

FLOW1(J) = FLOW(J)

ICOST(J) = 0

C UPDATE LINK COSTS, PROTECTING AGAINST ZERO-FLOW LINKS

IF(FLOW(J) .LE. 0) THEN

ICOST(J) = CO(J)

ELSE

ICOST(J)=CO(J)+C1(J)*((INT(REAL(FLOW(J)))*C2(J)))

ENDIF

FLOW(J) = 0

210 CONTINUE

C CALCULATE NEW LINK FLOWS BASED ON UPDATED COSTS

```

CALL MATRIX(ONODE,DNODE,ICOST,FLAG,NLINK,ORIGIN,DESTIN,FLOW,
-   FLOWIN,RESULT,TOTAL,KEY,NARC,MAXN,TCFLOW,G)

```

C FIND RELATIVE WEIGHTS OF FLOW AND FLOW1

CALL MSIZE(FLOW,FLOW1,ALPHA,CO,C1,C2,MSNO,NLINK,NARC)

DIFF = 0
FLOWS = 0

C CONVERGENCE TEST IS THE ROOT MEAN SQUARE OF THE FLOWS
DO 220 K=1,NARC

FLOW(K)=FLOW1(K) + (INT(ALPHA*(REAL(FLOW(K) - FLOW1(K))))
DIFF = DIFF + (FLOW1(K) - FLOW(K))*(FLOW1(K) - FLOW(K))
FLOWS = FLOWS + FLOW(K)

220 CONTINUE

IF(DIFF .LT. 0) DIFF = 32000
IF(FLOWS .EQ. 0) FLOWS = 1

TOTAL = (SQRT(REAL(DIFF)))/(REAL(FLOWS))

IF(TOTAL .LE. RESULT) GOTO 300

200 CONTINUE

300 DO 250 N = 1,NARC

C UPDATE LINK COSTS FOR ONE MORE ITERATION ON OUTPUT

IF(FLOW(N) .LE. 0) THEN
ICOST(N) = CO(N)

ELSE

ICOST(N) = CO(N) + C1(N)*(INT(REAL(FLOW(N))**C2(N)))

ENDIF

FLOW(N) = 0

250 CONTINUE

C KEY = 1 ALLOWS OUTPUT TO BE CALLED AT CONVERGENCE

KEY = 1

CALL MATRIX(ONODE,DNODE,ICOST,FLAG,NLINK,ORIGIN,DESTIN,FLOW,
- FLOWIN,RESULT,TOTAL,KEY,NARC,MAXN,TCFLOW,G)

RETURN

END

C GOLDEN SECTION METHOD IS USED TO FIND MOVE SIZE IN CONVEX

C COMBINATIONS PROCEDURE

SUBROUTINE MSIZE(FLOW,FLOW1,ALPHA,CO,C1,C2,MSNO,NLINK,NARC)
INTEGER MSNO,NLINK,NARC,FLOW(215),FLOW1(215),CO(300),C1(300)
REAL C2(300),XL,XR,ZL,ZR,R,A,B,ALPHA

A = 0.0

B = 1.0

R = 0.6180333

```

C  SET ITERATIONS COUNTER FROM INPUT
DO 10 N=1,MSNO

C  INITIALIZE PARAMETERS
    XL = (B-A)*(1-R) + A
    XR = (B-A)*R + A
    ZL = 0.0
    ZR = 0.0

    DO 7 I=1,NARC

C  CALCULATE OBJECTIVE FUNCTION AT LEFT AND RIGHT INTERVAL POINTS
    IF((FLOW1(I) +(INT((XL)*(REAL(FLOW(I) -
-      FLOW1(I)))))) .NE. 0) THEN
        ZL = ZL+(REAL(C1(I))*((REAL(FLOW1(I))) +
-      XL*(REAL(FLOW(I)-FLOW1(I))))**C2(I)
    ENDIF

    IF((FLOW1(I) +(INT((XR)*(REAL(FLOW(I) -
-      FLOW1(I)))))) .NE. 0) THEN
        ZR = ZR+(REAL(C1(I))*((REAL(FLOW1(I))) +
-      XR*(REAL(FLOW(I)-FLOW1(I))))**C2(I)
    ENDIF
7  CONTINUE

C  ELIMINATE INTERVAL FROM NEAREST END POINT TO POINT WITH HIGHER
C  OBJECTIVE FUNCTION VALUE
    IF(ZL .LE. ZR) THEN
        B = XR
    ELSE
        A = XL
    ENDIF

    IF( (B-A) .LE. 0.02 ) GOTO 30
10  CONTINUE

C  COMPUTE WEIGHT FACTOR AS THE MIDPOINT OF THE REMAINING INTERVAL
30  ALPHA = (A + B)/2.

    RETURN
    END

C  MATRIX USES REPEATED ITERATIONS OF A GREEDY ALGORITHM TO FIND THE
C  LOWEST COST PATH COST BETWEEN EVERY O-D PAIR.

    SUBROUTINE MATRIX (ONODE,DNODE,ICOST,FLAG,NLINK,ORIGIN,DESTIN,
-      FLOW,FLOWIN,RESULT,TOTAL,KEY,NARC,MAXN,TCFLOW,G)
    INTEGER ONODE(300),DNODE(300), FLAG(300),S(215),P(215),MINE(60),
-      KUTIL(215),ORIGIN(215),DESTIN(215),KEY,NARC,FLOW(215),
-      ICOST(215),L(215),FLOWIN(215),MAXN,TCFLOW,G

```

```

K4 = 0
K3 = 1
TCFLOW = 0

C EACH ORIGIN IS EVALUATED INDIVIDUALLY BUT THE LOWEST COST PATH TO
C EVERY DESTINATION IS FOUND ON A SINGLE ITERATION.
C FIND THE ORIGIN TO BE EVALUATED; FLAG = 1
  DO 500 K=1,NARC
  IF(FLAG(K) .NE. 1) GOTO 500
  MINE(K3) = ONODE(K)

C CHECK THAT AN ORIGIN IS VISITED ONLY ONCE
  IF(K3 .NE. 1) THEN
    DO 801 KM=1,(K3-1)
    IF(MINE(K3) .EQ. MINE(KM)) GOTO 500
801  CONTINUE
  ENDIF

C THE ARRAYS MUST BE REINITIALIZED FOR EACH ORIGIN
  DO 78 I=1,215
  S(I) = 0
  L(I) = 9999
  P(I) = 0
78  CONTINUE

C INITIALIZE THE ARRAYS FOR THE ORIGIN NODE; L IS THE DISTANCE FROM
C THE ORIGIN ON THE LOWEST COST PATH, S IS A LABELING ARRAY OF NODES
C TO BE EVALUATED AS ONODES.
  L(MINE(K3)) = 0
  S(MINE(K3)) = 1
  K3 = K3 + 1

  DO 514 J=1,NARC
  IF(S(J) .NE. 0) THEN
    DO 519 KA=K3-1,NARC

C FIND THE ONODE TO BE CONSIDERED, EVALUATE ALL DNODES OF LINKS FROM
C THAT ONODE FOR THE LOWEST COST PATH TO THE DNODE BY ANY ROUTE
    IF(ONODE(KA) .EQ. J) THEN
      IF(L(DNODE(KA)) .GT. L(ONODE(KA)) + ICOST(KA)) THEN
C UPDATE THE PATH COST AND PREDECESSOR ARRAYS IF LOWER COST PATH FOUND
        L(DNODE(KA)) = L(ONODE(KA)) + ICOST(KA)
        P(DNODE(KA)) = ONODE(KA)
      ENDIF

C ENTER NEW DNODES IN THE LABELING ARRAY
      S(DNODE(KA)) = 1
    ENDIF
519  CONTINUE

C REMOVE ONODE FROM LABELING ARRAY WHEN CONSIDERATION IS COMPLETED
  S(J) = 0
  ENDIF
514  CONTINUE

```



```

C OUTPUT HEADING ONLY FOR LAST ITERATION OF FLOW UPDATES AND ONLY
C THE FIRST ORIGIN
  IF(K4 .EQ. 0 .AND. KEY .EQ. 1) THEN
    K1 = 0
    DO 8010 M=1,NARC

C DETERMINE ALL DESTINATIONS IN ORDER AND ONLY ONE TIME FOR HEADING
    IF(FLAG(M) .EQ. 3 .AND. P(DNODE(M)) .EQ. ONODE(M)) THEN
      K1 = K1 + 1
      KUTIL(K1) = DNODE(M)
    ENDIF
8010    CONTINUE

    WRITE(*,2020)
    WRITE(6,2020)
2020    FORMAT(//////////,12X,'ORIGIN TO DESTINATION COST MATRIX')

    WRITE(*,2000) (KUTIL(M), M=1,K1)
    WRITE(6,2000) (KUTIL(M),M=1,K1)
2000    FORMAT(//,'UTILITIES',20I6,/)

    WRITE(*,2010)
    WRITE(6,2010)
2010    FORMAT('MINES')
      K4 =K4 +1
    ENDIF

    IF(KEY .EQ. 1) THEN

C OUTPUT IS CALLED TO OUTPUT PATH COSTS IN MATRIX FORMAT
      CALL OUTPUT(ONODE,DNODE,FLAG,NLINK,P,L,MINE,K3,NARC)

    ENDIF

C QSUM IS CALLED TO SUM THE FLOWS ON THE LINKS
      CALL QSUM2(ONODE,DNODE,FLAG,NLINK,P,MINE,K3,ORIGIN,DESTIN,FLOW,
-        FLOWIN,NARC,G)

500    CONTINUE

C TCFLOW REPRESENTS THE TOTAL SYSTEM COST OF THE EQUILIBRIUM FLOW PATTERN
      DO 750 L1 = 1,NARC
        TCFLOW = TCFLOW + (FLOW(L1) * ICOST(L1))
750    CONTINUE

      RETURN
      END

C OUTPUT IS USED TO WRITE THE PATH COSTS IN MATRIX FORMAT
      SUBROUTINE OUTPUT(ONODE,DNODE,FLAG,NLINK,P,L,MINE,K3,NARC)
      INTEGER ONODE(300),DNODE(300),FLAG(300),P(215),MINE(60),NARC,
-        L(215),NUTIL(215)

```

DATA NUTIL/215*0/
K2 = 0

DO 600 N=1,NARC

C DETERMINE DESTINATIONS AND BE CERTAIN TO VISIT EACH DESTINATION ONLY
C ONCE FROM EACH ORIGIN.

IF(FLAG(N) .EQ. 3 .AND. P(DNODE(N)) .EQ. ONODE(N)) THEN

K2 = K2 + 1

NUTIL(K2) = L(DNODE(N))

ENDIF

600 CONTINUE

WRITE(*,700)MINE(K3-1),(NUTIL(N1),N1=1,K2)

WRITE(6,700)MINE(K3-1),(NUTIL(N1),N1=1,K2)

700 FORMAT(I4,6X,24I6,/)

RETURN

END

C QSUM2 IS USED TO SUM THE FLOWS ON THE LINKS AS THE LOWEST COST PATHS
C ARE FOUND IN EACH ITERATION.

SUBROUTINE QSUM2(ONODE,DNODE,FLAG,NLINK,P,MINE,K3,ORIGIN,DESTIN,
- FLOW,FLOWIN,NARC,G)

INTEGER ONODE(300), DNODE(300), FLAG(300), NLINK,P(215),MINE(60)

- ,K3,PB(40),ORIGIN(215),DESTIN(215),NARC, FLOW(215),FLOWIN(215),G

DATA PB/40*0/

DO 800 N=1,NARC

KOUNT = 1

C DETERMINE THE PATH FROM THE DESTINATION TO THE ORIGIN USING THE
C PREDECESSOR ARRAY

IF(FLAG(N) .EQ. 3 .AND. P(DNODE(N)) .EQ. ONODE(N)) THEN

K8 = K8 + 1

PB(1) = DNODE(N)

DO 810 K=2,40

C CHECK FOR THE PROPER ORIGIN TO END THE PATH

IF(PB(K-1) .EQ. MINE(K3-1)) THEN

DO 900 M= 1,(KOUNT-1)

DO 920 I= 1,NARC

C CHECK THAT THE LINK IS ON THE PATH

IF(PB(M+1) .EQ. ONODE(I) .AND. PB(M).EQ.DNODE(I))THEN

DO 910 J= 1,G

```

C  CHECK THAT THE PROPER O-D FLOW IS ASSIGNED TO THE LINK
      IF(ORIGIN(J).EQ.MINE(K3-1) .AND. DESTIN(J).EQ.
-      DNODE(N)) THEN

C  ASSIGN THE FLOW TO EACH LINK ON THE PATH
      FLOW(I) = FLOW(I) + FLOWIN(J)
      GOTO 900

      910      ENDIF
              CONTINUE

      920      ENDIF
              CONTINUE
      900      CONTINUE

              GO TO 800

      ELSE

              PB(K) = P(PB(K-1))
              KOUNT = KOUNT + 1

      ENDIF

      810  CONTINUE

      ENDIF

      800  CONTINUE

      RETURN
      END

```

SUBROUTINE TRANS(H,DNODE,X,CPX,PN,S,CP,MAX,MAXN,NARC,ICOST)

C THIS SUBROUTINE CONTROLS THE TRANSSHIPMENT SEGMENT OF THE PROGRAM
C WHICH DETERMINES WHERE EACH UTILITY WILL BUY ITS COAL. IT TAKES
C AS INPUT, THE ROUTE STRUCTURE AND LINK COSTS AND FLOWS FROM THE
C FLOW DEPENDENT CODE CONTROLLED BY THE LCOST2 SUBROUTINE.

INTEGER H(125),DNODE(300),D(215),PN(215),IT(215),MAXN,PREV,
- NODEI,NODEJ,NODEC,NODED,INARC,PIVOT,IPIV,FPIV,TPIV,JOIN,
- HANG,NARC,ICOST(215),X(215),CPX(215),U(215),S(215),MAX,Z,
- CP(300),ZBEST,FCHG

LOGICAL FEAS

FEAS = .FALSE.

PIVOT = 0

IPIV = 0

FPIV = 0

TPIV = 0

FCHG = 0

NODEI = 0

NODEJ = 0

DO 50 J =1,215

D(J) = 0

PN(J) = 0

IT(J) = 0

X(J) = 0

U(J) = MAX

CPX(J) = 0

50 CONTINUE

C FORM THE INIRIAL SOLUTION FROM ALL THE ARTIFICIAL LINKS.
CALL INIT(MAXN,D,U,PN,H,X,CPX,DNODE,IT,PREV,MAX,S)

DO 10 I=1,500

IF(.NOT. FEAS) THEN

C CHOOSE LINK WHICH WILL MOST IMPROVE THE SOLUTION

CALL ARCIN(H,DNODE,ICOST,MAXN,CP,U,ZBEST,NODEI,NODEJ,INARC
- ,FEAS,PIVOT)

IF(.NOT. FEAS) THEN

C DETERMINE WHAT ACTION TO TAKE WHEN EACH NEW LINK IS ADDED TO THE
C SOLUTION

CALL ARCOUT(ZBEST,D,X,CPX,PN,NODEI,NODEJ,NODEC,NODED,
- HANG,FCHG,U,IT,INARC,ICOST,CP,MAX,JOIN,H,DNODE,
- IPIV,FPIV,TPIV)

```

        ENDIF

        ELSE
            GOTO 20
        ENDIF

10  CONTINUE
20  CONTINUE

    WRITE(6,200) PIVOT,IPIV,FPIV,TPIV
200  FORMAT(/,' PIVOT = ',I3,2X,' IPIV = ',I3,2X,' FPIV = ',I3,2X,
-      ' TPIV = ',I3)

    RETURN
    END

C  THIS SUBROUTINE FORMS THE INITIAL SOLUTION FROM ALL THE ARTIFICIAL LINKS

    SUBROUTINE INIT(MAXN,D,U,PN,H,X,CPX,DNODE,IT,PREV,MAX,S)
    INTEGER D(215),PN(215),MAXN,PREV,IT(215),H(125),DNODE(300),NODE,
-      S(215),X(215),CPX(215),MAX,U(215)

C  INITIALIZE THE NODE LENGTH ARRAYS FOR THE ARTIFICIAL NODE WHICH IS
C  THE ROOT OF THE SOLUTION TREE
    D(MAXN) = 1
    U(MAXN) = 0
    PN(MAXN) = MAXN
    PREV = MAXN

    DO 20 I=1,MAXN
        IF(I .GE. MAXN) GOTO 30
C  FOR ALL THE SUPPLY NODES (S>0), INITIALIZE THE ARRAYS WITH X CARRYING
C  THE FLOW
        IF(S(I) .GT. 0) THEN
            X(I) = S(I)
            CPX(I) = MAX - S(I)
            D(I) = 2
            U(I) = 0
            PN(I) = MAXN
            IT(PREV) = I
            PREV = I
        ENDIF
20  CONTINUE
30  I = I - 1
    J = H(MAXN)

```

```

DO 40 K=1,MAXN
C  FOR ALL THE LINKS FROM THE ARTIFICIAL NODE, CPX WILL CARRY THE FLOW
C  INITIALIZE ALL THE NODE ARRAYS HERE
      IF(J .LE. (H(MAXN + 1) - 1)) THEN
        NODE = DNODE(J)
        X(NODE) = MAX + S(NODE)
        CPX(NODE) = - S(NODE)
        D(NODE) = 2
        U(NODE) = MAX-1
        PN(NODE) = -MAXN
        IT(PREV) = NODE
        PREV = NODE
        J = J + 1

      ELSE
        GOTO 50
      ENDIF
40  CONTINUE

C  COMPLETE PREORDER TRAVERSAL; RETURN TO THE ARTIFICIAL NODE.
50  IT(PREV) = MAXN

      RETURN
      END

```

```

C  THIS SUBROUTINE FINDS THE LINK WITH THE GREATEST DUAL INFEASIBILITY
C  AND DESIGNATES THAT LINK AS THE CANDIDATE INCOMING LINK

      SUBROUTINE ARCIN(H,DNODE,ICOST,MAXN,CP,U,ZBEST,NODEI,NODEJ,INARC,
-      FEAS,PIVOT)
      INTEGER MAXN,NODEI,NODEJ,H(125),DNODE(300),INARC,PIVOT,
-      ICOST(215),Z,U(215),ZBEST,CP(300),J,K
      LOGICAL FEAS

C  ZBEST REGISTERS THE GREATEST DUAL INFEASIBILITY FROM NODEI WHICH IS
C  BEING INVESTIGATED
      ZBEST = 0
      DO 110 I=1,MAXN
        IF(ZBEST .EQ. 0 .AND. I .LE. MAXN) THEN
          NODEI = MOD(NODEI,MAXN) + 1
          K = H(NODEI)
          DO 120 J = 1,500
            IF(K .LE. (H(NODEI + 1) - 1)) THEN
              Z = (U(DNODE(K)) - U(NODEI)) - ICOST(K)

              IF((CP(K) .GT. 0 .AND. Z .GT. IABS(ZBEST)) .OR.
-              (CP(K) .LT. 0 .AND. -Z .GT. IABS(ZBEST))) THEN

```

```

        ZBEST = Z
        INARC = K
    ENDIF
    K = K + 1
ELSE
    GOTO 110
ENDIF
120 CONTINUE
ELSE
    GOTO 130
ENDIF
110 CONTINUE

C IF ZBEST = 0 AT THIS POINT WE ARE FINISHED.  NODEI-NODEJ DESIGNATE THE
C INCOMING ARC IF ZBEST IS NOT ZERO.
130 NODEJ = DNODE(INARC)

    IF(ZBEST .EQ. 0) FEAS = .TRUE.

    PIVOT = PIVOT + 1

    RETURN
    END

```

C ARCOUT DETERMINES WHAT TO DO WHEN A LINK ENTERS THE SOLUTION

```

    SUBROUTINE ARCOUT(ZBEST,D,X,CPX,PN,NODEI,NODEJ,NODEC,NODED,
-   HANG, FCHG,U,IT,INARC,ICOST,CP,MAX,JOIN,H,DNODE,IPIV,FPIV,
-   TPIV)
    INTEGER ANODE,BNODE,D(215),PN(215),NODEI,NODEJ,NODEC,NODED,
-   HANG,IT(215),INARC,JOIN,H(125),DNODE(300),IPIV,FPIV,TPIV,
-   ZBEST,FCHG,X(215),CPX(215),U(215),ICOST(215),CP(300),MAX

```

C DETERMINE IF EITHER NODE IS BEING REACHED FOR THE FIRST TIME.

```

    IF(PN(NODEJ) .EQ. 0 .OR. PN(NODEI) .EQ. 0) THEN

```

C IF A NEW NODE IS REACHED, THE NODE ARRAYS MUST BE INITIALIZED.

```

    CALL INODE(U,D,PN,X,CPX,IT,NODEI,NODEJ,INARC,ICOST,CP,MAX,
-   IPIV)

```

```

    ELSE

```

C CHECK TO SEE WHICH SIDE OF THE TREE HOLDS THE FLOW. ANODE IS THE LEFT
C SIDE OF THE TREE WHERE X HOLDS THE FLOW. IF INCOMING LINK IS AT CAPACITY
C EXPECT A FLOW DECREASE; OTHERWISE A FLOW INCREASE.

```

    IF(ZBEST .LT. 0) THEN

```

```

        ANODE = NODEJ

```

```

        BNODE = NODEI

```

```

    ELSE

```

```

        ANODE = NODEI

```

```

        BNODE = NODEJ

```

```

    ENDIF

```

```

FCHG = MAX

C SYNCHRONOUSLY ITERATE ANODE AND BNODE BACK TO THEIR JOIN, KEEPING TRACK
C OF THE MAX FLOWCHANGE AND THE SIDE OF THE TREE WHERE IT OCCURS.
DO 10 K=1,500
  IF( ANODE .NE. BNODE .AND. FCHG .GT. 0) THEN
    IF(D(ANODE) .GE. D(BNODE)) THEN
      IF(X(ANODE) .LT. FCHG) THEN
        FCHG = X(ANODE)
        NODEC = ANODE
      ENDIF
      ANODE = IABS(PN(ANODE))
    ENDIF

    IF(D(BNODE) .GT. D(ANODE)) THEN
      IF(CPX(BNODE) .LT. FCHG) THEN
        FCHG = CPX(BNODE)
        NODEC = - BNODE
      ENDIF
      BNODE = IABS(PN(BNODE))
    ENDIF

    ELSE
      GOTO 101
    ENDIF
  10 CONTINUE

C IF MINIMUM FLOW OCCURS ON CPX SIDE, THEN HANGNODE = NODEJ; ELSE
C HANGNODE = NODEI
101 IF((ISIGN(1,ZBEST)*ISIGN(1,NODEC)) .GT. 0) THEN
  HANG = NODEJ
ELSE
  HANG = NODEI
ENDIF

C RESTORE PLUS SIGN TO NODEC; SAVE PREDECESSOR OF NODEC AND JOIN
NODEC = IABS(NODEC)
NODED = IABS(PN(NODEC))
JOIN = ANODE

IF(FCHG .NE. 0) CALL FLOWUP(ZBEST,FCHG,CP,INARC,NODEI,JOIN,
- X,CPX,FPIV,PN,NODEJ)

IF(IABS(FCHG) .LT. IABS(CP(INARC))) CALL TREEUP(PN,NODEC,
- H,DNODE,CP,HANG,NODEI,NODEJ,ZBEST,INARC,FCHG,IT,D,X,
- TPIV,NODED,CPX,U)

ENDIF

RETURN
END

```



```

SUBROUTINE INODE(U,D,PN,X,CPX,IT,NODEI,NODEJ,INARC,ICOST,CP,MAX,
- IPIV)
  INTEGER D(215),PN(215),IT(215),NODEI,NODEJ,INARC,IPIV,
- U(215),X(215),CPX(215),ICOST(215),CP(300),MAX

```

C WHEN NODE IS REACHED FOR THE FIRST TIME, THIS SUBROUTINE UPDATES THE
 C NODE ARRAYS. LINK ENTERS A ZERO FLOW, AND SINCE NO BACKPATH EXISTS,
 C THIS IS THE END OF THIS IMPROVEMENT CYCLE.

```

  IF(U(NODEI) .EQ. MAX) THEN
    D(NODEI) = D(NODEJ) + 1
    PN(NODEI) = NODEJ
    X(NODEI) = 0
    CPX(NODEI) = CP(INARC)
    U(NODEI) = U(NODEJ) - ICOST(INARC)
    IT(NODEI) = IT(NODEJ)
    IT(NODEJ) = NODEI

```

```

  ELSE
    D(NODEJ) = D(NODEI) + 1
    PN(NODEJ) = - NODEI
    X(NODEJ) = CP(INARC)
    CPX(NODEJ) = 0
    U(NODEJ) = U(NODEI) + ICOST(INARC)
    IT(NODEJ) = IT(NODEI)
    IT(NODEI) = NODEJ

```

```

ENDIF

```

```

  IPIV = IPIV + 1

```

```

RETURN
END

```

C FLOW IS UPDATED WHETHER THE LINK ENTERS THE SOLUTION OR NOT. IF LINK
 C DOES NOT ENTER, THIS ENDS THE CYCLE. IF LINK DOES ENTER THE SOLUTION
 C THEN TREEUP WILL BE CALLED TO UPDATE ALL THE NODE ARRAYS

```

SUBROUTINE FLOWUP(ZBEST,FCHG,CP,INARC,NODEI,JOIN,X,CPX,FPIV,PN,
- NODEJ)
  INTEGER NODE,INARC,NODEI,DIR,JOIN,FPIV,PN(215),NODEJ,
- ZBEST,CP(300),X(215),CPX(215),FCHG

```

C IF Z<0, THE LINK IS AT CAPACITY AND A FLOW REDUCTION WILL OCCUR TO
 C ENTER THE BASIS. FLOWCHANGE WILL BE THE MINIMUM ALLOWED BY INCOMING
 C LINK AND BACKPATH.

```

IF(ZBEST .LT. 0) THEN
  FCHG = MAX0(CP(INARC),(-FCHG))
ELSE
  FCHG = MIN0(CP(INARC),FCHG)
ENDIF

```

```

C IF FLOWCHANGE = CAPACITY, THE LINK WILL NOT ENTER THE BASIS. SIGN
C CHANGED ON CAPACITY AND FLOWS ARE UPDATED TO END CYCLE.

```

```

IF(FCHG .EQ. CP(INARC)) THEN
  CP(INARC) = - CP(INARC)
ENDIF

```

```

DIR = 1
NODE = NODEI

```

```

C ITERATE UP BOTH ARMS OF THE BACKPATH ADJUSTING FLOW.

```

```

DO 100 I=1,2
  DO 200 J=1,500
    IF(NODE .NE. JOIN) THEN
      X(NODE) = X(NODE) - (FCHG * DIR)
      CPX(NODE) = CPX(NODE) + (FCHG * DIR)
      NODE = IABS(PN(NODE))
    ELSE
      GOTO 300
    ENDIF
  200 CONTINUE
  300 NODE = NODEJ
  DIR = -1
  100 CONTINUE

  FPIV = FPIV + 1

  RETURN
END

```

```

SUBROUTINE TREEUP(PN,NODEC,H,DNODE,CP,HANG,NODEI,NODEJ,
- ZBEST,INARC,FCHG,IT,D,X,TPIV,NODED,CPX,U)

```

```

C THIS SUBROUTINE IS THE MAJOR PART OF THE PIVOT, UPDATING THE
C NODE LENGTH ARRAYS X,CPX,D,U,PN,IT
  INTEGER PN(215),H(125),IT(215),DNODE(300),D(215),PF,NR,NC,NT,LR,
- ORIG,DADJ,NODE,NODE2,NODEC,NODED,NODEI,NODEJ,HANG,
- INARC,LASTP,TPIV,
- CP(300),X(215),CPX(215),FLO,ZBEST,FCHG,LASTX,LASTC,U(215)
  LOGICAL LAST

```

C MARK THE OUTGOING LINK; FIND ITS ORIENTATION AND LOCATE ITS HEAD
 IF(PN(NODEC) .GT. 0) THEN

 NODE = NODEC
 NODE2 = NODED
 FLO = X(NODEC)
 ELSE
 NODE = NODED
 NODE2 = NODEC
 FLO = CPX(NODEC)
 ENDIF

C ITERATE THROUGH DNODES UNTIL LINK IS FOUND

 J = H(NODE)
 DO 70 K=1,500
 IF(J .LE. (H(NODE + 1) - 1)) THEN
 IF(DNODE(J) .EQ. NODE2) THEN
 IF(FLO .EQ. IABS(CP(J))) THEN
 CP(J) = - IABS(CP(J))
 ELSEIF (FLO .EQ. 0) THEN
 CP(J) = IABS(CP(J))
 ELSE
 WRITE(6,71)NODEC,NODED,FLO,TPIV
 FORMAT(/,4X,' CP MARKING ERROR: ',4I10)
 ENDIF
 GOTO 20
 ENDIF
 J = J + 1
 ELSE
 GOTO 20
 ENDIF
 71
 CONTINUE
 20 CONTINUE

C CHECK ORIENTATION OF INCOMING LINK BY CHEKING ITS HANGNODE. THEN
 C INITIALIZE THE PIVOT VARIABLES.

 IF(HANG .EQ. NODEI) THEN
 PF = -NODEI
 NR = NODEJ
 ZBEST = - ZBEST
 ELSE
 PF = NODEJ
 NR = NODEI
 ENDIF

C THERE ARE FOUR POSSIBLE COMBINATIONS OF LINK ORIENTATION AND FLOW
 C DIRECTION. BY CHECKING SIGNS OF PF AND ZBEST THESE FOUR ARE CAPTURED.

 IF(ZBEST .GT. 0) THEN
 LASTC = IABS(CP(INARC) - FCHG)
 LASTX = IABS(FCHG)

```

ELSE
  LASTC = IABS(FCHG)
  LASTX = IABS(CP(INARC) - FCHG)
ENDIF

```

```

C INITIALIZE THE REMAINING PIVOT TREE VARIABLES, KEEPING TRACK OF THREAD
C FROM HANGNODE.

```

```

  NC = IABS(PF)
  LR = IT(NR)
  NT = LR
  DADJ = D(NC) - D(NR) - 1
  ORIG = IT(NC)
  NODE = NODED

```

```

C FIND NODE S.T. IT(NODE) = NODEC AND CHANGE ITS IT TO THAT OF THE LAST
C RIGHR SUCCESSOR OF NODEC.

```

```

  DO 80 M = 1,500
    IF(IT(NODE) .NE. NODEC) THEN
      NODE = IT( NODE)
    ELSE
      GOTO 81
    ENDIF

```

```

80 CONTINUE

```

```

81 LAST = .FALSE.

```

```

  DO 90 M2 = 1,500
    IF(.NOT. LAST) THEN
      LAST = (NR .EQ. NODEC)
      IT(NC) = NR
      NC = NR
      DADJ = DADJ + 2
      DO 91 M3 = 1,500
        IF(IT(NC) .NE. LR) THEN
          NC = IT(NC)
          D(NC) = D(NC) + DADJ
          U(NC) = U(NC) + ZBEST
        ELSE
          GO TO 99
        ENDIF
      CONTINUE

```

```

99 IF(D(NT) .GT. D(NR)) THEN

```

```

  IT(NC) = NT
  DO 92 M4 = 1,500
    IF(D(NT) .GT. D(NR)) THEN
      NC = NT
      NT = IT(NT)
      D(NC) = D(NC) + DADJ
      U(NC) = U(NC) + ZBEST
    ELSE
      GOTO 98
    ENDIF
  CONTINUE
98 ENDIF

```

```

92

```

```

98

```

```

        LR = NR
        NR = IABS(PN(LR))
        LASTP = PN(LR)
        PN(LR) = PF
        D(LR) = D(LR) + DADJ
        U(LR) = U(LR) + ZBEST
        PF = - ISIGN(LR, LASTP)

    ELSE
        GOTO 97
    ENDIF
90  CONTINUE

C  COMPLETE THREAD UPDATE; IF NODEC'S THREAD PREDECESSOR IS THE HANGNODE,
C  THEN ONLY ONE EXTRA CONNECTION NEEDS TO BE MADE; OTHERWISE BOTH NODES
C  MUST BE CONNECTED.

97  IF(NODE .EQ. HANG) THEN
        IT(NC) = NT
    ELSE
        IT(NC) = ORIG
        IT(NODE) = NT
    ENDIF

C  UPDATE X AND CPX ALONG THE PIVOT STEM, WHICH IS TURNED UPSIDE DOWN.
C  THIS STEP IS MANIPULATING THESE FLOWS WHICH HAVE ALREADY BEEN UPDATED
C  IN FLOWUP SO THAT THEY CORRESPOND TO CHANGES IN THE TREE.
    NODE = LR
    DO 100 L = 1, 500
        IF(IABS(PN(NODE)) .NE. HANG) THEN
            X(NODE) = CPX(IABS(PN(NODE)))
            CPX(NODE) = X(IABS(PN(NODE)))
            NODE = IABS(PN(NODE))
        ELSE
            GOTO 101
        ENDIF
    100 CONTINUE

C  ENTER X AND CPX OF THE INCOMING LINK.
101  X(NODE) = LASTX
    CPX(NODE) = LASTC
    TPIV = TPIV + 1
    RETURN
    END

```

C THIS SUBROUTINE IS USED TO OUTPUT THE LINK FLOW AND COST DATA FROM
 C BOTH THE FLOW-DEPENDENT AND THE TRANSSHIPMENT ALGORITHMS.

```

    SUBROUTINE NETOUT(ONODE,DNODE,ICOST,FLOW,PN,X,CPX,KEY,NARC,NNODE,
-      MAXN,TCFLOW,CP)
    INTEGER PN(215),DNODE(300),NARC,ICOST(215),X(215),ONODE(300),
-      FLOW(215),CPX(215),KEY,NNODE,MAXN,TCFLOW,CP(300),TCTRAN
  
```

C KEY = 1 DESIGNATES THAT OUTPUT IS FROM THE FLOW-DEPENDENT ALGORITHM
 K10 = 0

```

    IF(KEY .EQ. 1) THEN
  
```

C WRITE THE HEADING, NOTE THAT THE OUTPUT INCLUDES THE FLOW AND THE COST
 C FOR EVERY LINK IN THE NETWORK. THESE ARE THE EQUILIBRIUM FLOWS AND
 C COSTS ON THE LINKS. THE PATH COSTS HAVE BEEN GIVEN ALREADY FROM MATRIX.

```

        WRITE(*,5)
        WRITE(6,5)
    5    FORMAT(/////,20X,'LINK FLOW & COST DATA',/,18X,
-      '(FLOW-DEPENDENT ALGORITHM)',///,11X,'ORIGIN',7X,
-      'DESTINATION',3X,'AMOUNT OF',9X,'LINK',/,12X,
-      'NODE',11X,'NODE',9X,'FLOW',12X,'COST',//)
  
```

```

        DO 10 N = 1,NARC
  
```

C OUTPUT IS GIVEN ONLY FOR THE REAL LINKS IN THE NETWORK. MAXN DESIGNATES
 C THE ARTIFICIAL NODE, AND LINKS TO THAT NODE ARE EXCLUDED.

```

        IF(ONODE(N) .NE. MAXN .AND. DNODE(N) .NE. MAXN) THEN
            WRITE(*,12) ONODE(N),DNODE(N),FLOW(N),ICOST(N)
            WRITE(6,12) ONODE(N),DNODE(N),FLOW(N),ICOST(N)
    12    FORMAT(4I15)
        ENDIF
    10    CONTINUE
  
```

C TCFLOW IS THE TOTAL COST GIVEN BY THE FLOW-DEPENDENT EQUILIBRIUM FLOW
 C PATTERN.

```

        WRITE(*,11) TCFLOW
        WRITE(6,11) TCFLOW
    11    FORMAT(/////,7X,'FLOW-DEPENDENT ALGORITHM TOTAL COST = ',18,//)

        ENDIF

        TCTRAN = 0
  
```

C KEY = 2 DESIGNATES THE TRANSSHIPMENT ALGORITHM OUTPUT.

IF(KEY .EQ. 2) THEN

C FOR THE TRANSSHIPMENT ALGORITHM, EVERY NODE IN THE SOLUTION INITIATES
C ONE LINK TO ITS PREDECESSOR. THESE ARE THE ONLY LINKS LISTED, AND ONLY
C THE FLOW IS GIVEN SINCE THE COST IS THE SAME AS THE FLOW-DEPENDENT
C LINK COSTS ABOVE.

```
      WRITE(*,15)
      WRITE(6,15)
15    FORMAT(/////,20X,'LINK FLOW DATA',/,14X,'(TRANSSHIPMENT '
-      'ALGORITHM)',///,11X,'ORIGIN',7X,'DESTINATION',3X,
-      'AMOUNT OF',/,12X,'NODE',11X,'NODE',9X,'FLOW',//)
```

DO 50 I = 1,NARC

IF(PN(I) .GT. 0) THEN

C OUTPUT GIVEN IN THE SAME ONODE-DNODE-FLOW FORMAT AS FOR THE FLOW-
C DEPENDENT OUTPUT. IF PN(I) > 0, THE LINK IS ORIENTED AS DESIGNATED.
C IF PN(I) < 0, THE LINK IS THE BACKLINK OF THE NODE BEING EXAMINED.
C OUTPUT HAS BEEN WRITTEN TO SIMPLIFY ITS USE.

```
      WRITE(*,20) I,PN(I),X(I)
      WRITE(6,20) I,PN(I),X(I)
```

DO 90 K7 = 1,NARC

```
      IF(I .EQ. ONODE(K7) .AND. PN(I) .EQ. DNODE(K7)) THEN
        TCTRAN = TCTRAN + ICOST(K7) * X(I)
```

```
      ENDIF
90    CONTINUE
```

ELSEIF(PN(I) .LT. 0) THEN

```
      WRITE(*,20) IABS(PN(I)),I,CPX(I)
      WRITE(6,20) IABS(PN(I)),I,CPX(I)
```

DO 91 K8 = 1,NARC

```
      IF(PN(I) .EQ. ONODE(K8) .AND. I .EQ. DNODE(K8)) THEN
        TCTRAN = TCTRAN + ICOST(K8) * CPX(I)
```

```
      ENDIF
91    CONTINUE
```

ENDIF

```
20      FORMAT(3I15)
50    CONTINUE
```

```

DO 92 K9 = 1,NARC
  IF(CP(K9) .EQ. -IABS(CP(K9))) THEN
    IF(K10 .EQ. 0) THEN
      WRITE(6,89)
89      FORMAT(///,17X,'CAPACITATED LINKS')
    ENDIF

    IF(PN(K9) .EQ. 0) THEN
      TCTRAN = TCTRAN + ICOST(K9) * IABS(CP(K9))
    ENDIF

    WRITE(6,88) ONODE(K9),DNODE(K9),IABS(CP(K9))
88      FORMAT(3I15)
      K10 = K10 + 1
    ENDIF
92 CONTINUE

  IF(K10 .EQ. 0) THEN
    WRITE(6,87)
87      FORMAT(12X, 'THERE ARE NO LINKS CARRYING FLOW AT CAPACITY')
  ENDIF

  WRITE(*,95) TCTRAN
  WRITE(6,95) TCTRAN
95  FORMAT(//////,7X,'TRANSSHIPMENT ALGORITHM TOTAL COST = ',I8,/)

  ENDIF

  RETURN
  END

```


C THIS SUBROUTINE INVESTIGATES THE LOWEST COST PATH FROM ONE ORIGIN
C TO ON DESTINATION USING CONSTANT LINK COSTS.

```

SUBROUTINE SINGL3(ONODE,DNODE,ICOST,FLAG,NLINK,NARC)
  INTEGER MINE1,UTIL,ONODE(300),DNODE(300),FLAG(300),SI(215),
  - P(215),PA(215),NARC,ICOST(215),L(215)

```

C INPUT BY USER IS REQUIRED. ORIGIN AND DESTINATION MUST AGREE WITH
C NETWORK STRUCTURE INPUT.

```

DATA SI,L,P,PA/215*0,215*9999,215*0,215*0/
DATA MINE1,UTIL/0,0/
KOUNT = 1

```

```

WRITE(*,1030)
WRITE(6,1030)
1030 FORMAT(20X,'MINE OPTIONS',//,2X,'CHOOSE THE ORIGIN MINE FROM THE'
- ' LIST OF ORIGIN MINES ON THE FIRST MENU',/,8X,'(NOTE: FOR ANY'
- ' LINK WHERE FLAG = 1, THE ONODE REPRESENTS AN ORIGIN MINE)',/,
- 2X,'ENTER THE ONODE NUMBER OF YOUR CHOSEN MINE',/)
READ(*,1040) MINE1
WRITE(6,1040) MINE1
1040 FORMAT(BN,I3)

WRITE(*,1050)
WRITE(6,1050)
1050 FORMAT(/,20X,'UTILITY DESTINATION CHOICES',//,2X,'CHOOSE THE '
- 'DESTINATION UTILITY FROM THE LIST ON THE LAST INPUT MENU',/,
- 8X,'(NOTE: FOR ANY LINK WHERE FLAG = 3, THE DNODE REPRESENTS'
- ' A DESTINATION UTILITY)',/,2X,'ENTER THE DNODE NUMBER OF '
- 'YOUR CHOSEN UTILITY',/)
READ(*,1060) UTIL
WRITE(6,1060) UTIL
1060 FORMAT(BN,I3)

```

C THIS IS THE SAME GREEDY ALGORITHM AS USED IN MATRIX, BUT RUN ONLY ONCE
C HERE.

C L= PATH LENGTH TO A GIVEN NODE ON LOWEST COST PATH
C SI= LABE VARIABLE - DESIGNATES NODES TO INVESTIGATE AS ONODES

```

L(MINE1) = 0
SI(MINE1) = 1
DO 14 J=1,NARC

```

C FIND NEXT NODE TO BE INVESTIGATED AS AN ONODE

```

IF(SI(J) .NE. 0) THEN
  DO 19 K=1,NARC

```

C INVESTIGATE ALL LINKS FROM THE NODE - UPDATE PATH COST AND PREDECESSOR
C AS REQUIRED.

```

      IF(ONODE(K) .EQ. J) THEN
        IF(L(DNODE(K)) .GT. L(ONODE(K)) + ICOST(K)) THEN
          L(DNODE(K)) = L(ONODE(K)) + ICOST(K)
          P(DNODE(K)) = ONODE(K)
        ENDIF
      ENDIF

```

C DESIGNATE ALL DNODES REACHED AS POTENTIAL ONODES

```

      SI(DNODE(K)) = 1
    ENDIF
19    CONTINUE
      SI(J) = 0
    ENDIF
14    CONTINUE

```

C WHEN COST TO DESTINATION FOUND, RECONSTRUCT PATH BACK TO THE ORIGIN
C USING PREDECESSOR ARRAY.

```

      PA(1) = P(UTIL)
      DO 205 K=2,NARC
        IF(PA(K-1) .EQ. MINE1) THEN

```

C OUTPUT PATH BY LINK AND GIVE TOTAL PATH COST.

```

      WRITE(*,200)MINE1,UTIL,L(UTIL),(PA(J),J=KOUNT-1,1,-1)
      WRITE(6,200)MINE1,UTIL,L(UTIL),(PA(J),J=KOUNT-1,1,-1)
200    FORMAT(/,'THE MINIMUM COST FROM MINE # ',I3,' TO UTILITY '
- '# ',I3,' IS',I8,' AND THE ROUTE PASSES THROUGH NODES ',
- 50(I3,', '))
      GOTO 2
    ELSE
      PA(K) = P(PA(K-1))
      KOUNT = KOUNT + 1
    ENDIF
205    CONTINUE

```

C IF NO PATH EXISTS, THE L VALUE WILL REMAIN AS INITIALIZED.

```

      IF (L(UTIL) .EQ. 9999) THEN
        WRITE(*,206) MINE1,UTIL
        WRITE(6,206) MINE1,UTIL
206    FORMAT(3X,' THERE IS NO PATH FROM MINE ',I3,' TO UTILITY ',I5)
      ENDIF

2    RETURN
END

```

END

FILMED

10-84

DTIC